

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ESAT
AFDELING PSI
Kardinaal Mercierlaan 94 — 3001 Heverlee, Belgium

LOCAL, INVARIANT FEATURES FOR REGISTRATION AND RECOGNITION

Promotor:
Prof. L. VAN GOOL

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de Toegepaste Wetenschappen

door

Tinne TUYTELAARS

December 2000



KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ESAT
AFDELING PSI
Kardinaal Mercierlaan 94 — 3001 Heverlee, Belgium

LOCAL, INVARIANT FEATURES FOR REGISTRATION AND RECOGNITION

Jury:
Voorzitter:
Prof. L. Van Gool, promotor
Prof. H. Bruyninckx
Prof. S. Carlsson (KTH, Stockholm)
Prof. T. Moons (KUB, Brussel)
Prof. L. Van Eycken
Prof. P. Wambacq

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de Toegepaste Wetenschappen

door

Tinne TUYTELAARS

U.D.C. ???.?*???

December 2000

©Katholieke Universiteit Leuven - Faculteit Toegepaste Wetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

Wettelijk depot D/2000/..../..

ISBN ..-....-...-.

Acknowledgements

This dissertation would have been impossible without the encouragement and support of countless people over the past four years. I cannot possibly thank everyone individually, but would like to express my gratitude to them all.

Firstly, I'd like to thank my supervisor, Prof. Luc Van Gool, who gave me the opportunity to work in his research group. I'm especially thankful to him for his help with resolving theoretical problems and writing papers and for keeping me on the right track during the course of my PhD.

I am grateful to the members of my reading committee, Prof. Herman Bruyninckx, Prof. Luc Van Eycken and Prof. Patrick Wambacq, who all carefully read the manuscript in just a few weeks time, for their comments, suggestions and helpful feedback. I am also grateful to the other members of the jury, Prof. Stefan Carlsson and Prof. Theo Moons, who accepted this task with enthusiasm.

Next, I'd like to express my gratitude to my friends and colleagues, both past and present, from the VISICS Lab, for sharing their opinions in many interesting technical and non-technical discussions, for warm personal contact and an enjoyable atmosphere. They each helped make the past four years at the university more fun and interesting.

I'm also very grateful to the Fund for Scientific Research Flanders, who has made this research possible by providing a fellowship, leaving me free of financial or administrative burdens.

Finally, I am also indebted to my parents who encouraged and supported me from the very beginning, and last but not least to Pascal, who always knew exactly when to push me and when to tone me down.

Abstract

This dissertation deals with the problem of finding correspondences between two views of the same object or scene, taken from substantially different viewing angles, usually referred to as *wide baseline stereo*. The proposed solution is based on the notion of *local, affinely invariant regions*. These are small, viewpoint dependent parallelogram or ellipse-shaped image patches that, when changing the position of the camera, change their shape in the image accordingly in order to keep on representing the same physical part of the object or scene. Also photometric deformations due to changing lighting conditions are fully taken into account. Applications range from object recognition, over content-based image retrieval to vision-based navigation for an intelligent mobile robot.

In deze verhandeling wordt het bepalen van correspondenties tussen twee beelden van eenzelfde voorwerp of scène, genomen vanuit sterk verschillende gezichtspunten, bestudeerd. De voorgestelde oplossing is gebaseerd op de idee van *lokale, affien invariante omgevingen*. Dit zijn kleine, gezichtspuntsafhankelijke, parallellogram- of ellipsvormige omgevingen in het beeld die bij wijzigend gezichtspunt hun vorm zodanig aanpassen (op basis van het onderliggende intensiteitsprofiel) dat ze steeds hetzelfde fysische deel van het voorwerp of de scène beschrijven. Ook fotometrische variaties, ten gevolge van veranderende belichtingsomstandigheden, worden mee in rekening gebracht. Verschillende toepassingen werden uitgewerkt, gaande van voorwerpsherkenning, over het zoeken naar gelijkaardige beelden in een databank op basis van de beeldinhoud, tot visie-gebaseerde navigatie voor een intelligente mobiele robot.

Notations

To enhance the readability some notations and naming conventions used throughout the text are shortly summarized here.

For matrices bold face fonts are used (e.g. $\mathbf{A}$). 4-vectors are represented by $\mathtt{A}$ and 3-vectors by $\mathtt{a}$. Scalar values will be represented as a. $\mathbf{A_{ij}}$ refers to the element of matrix $\mathbf{A}$ on the ith row and jth column.

$\sim$	equivalence up to scale ($A \sim B \Leftrightarrow \exists \lambda \neq 0 : A = \lambda B$)
$\lvert\mathbf{A}\rvert$	determinant of matrix $\mathbf{A}$
$\mathbf{A}^T$	transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$ (i.e. $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$)
$\lVert\mathtt{v}\rVert$	the 2-norm of a vector ($\lVert\mathtt{v}\rVert = \sqrt{\sum_i v_i^2}$)

Moreover, the following symbols are reserved for the following entities:

$\mathtt{P}$	world point (4-vector)
$\mathtt{p}$	image point (3-vector)
$\mathbf{H}$	homography (3×3 matrix)
$\mathbf{F}$	fundamental matrix (3×3 rank 2 matrix)
$\mathbf{E}$	essential matrix (3×3 rank 2 matrix)
$\mathtt{e}$	epipole
$\mathbf{K}$	calibration matrix (3×3 upper triangular matrix)
$\mathbf{R}$	rotation matrix
$\mathtt{t}$	translation vector
$\mathbf{I}$	the identity matrix
$I(x,y)$	image intensity (in case of color images: either of the three colorbands R, G or B)
$\widehat{I}$	average image intensity
M_{pq}^{abc}	generalized color moment ($M_{pq}^{abc} = \iint_{\Omega} x^p y^q [R(x,y)]^a [G(x,y)]^b [B(x,y)]^c \, dxdy$)

Contents

1 Introduction **1**
- 1.1 Finding correspondences between images . . . 2
- 1.2 Recognizing objects from a limited number of reference views . 2
- 1.3 Other applications . . . 3
- 1.4 Main contributions . . . 5
- 1.5 Outline of the thesis . . . 6

2 State of the art **9**
- 2.1 Related work . . . 9
  - 2.1.1 Model-based versus appearance-based object recognition 9
  - 2.1.2 Object recognition techniques based on matching local features . . . 12
  - 2.1.3 Object recognition techniques based on statistics of local features . . . 15
  - 2.1.4 Increasing the robustness to viewpoint changes . . . 16
  - 2.1.5 The approach of Schmid and Mohr . . . 18
- 2.2 Conclusion . . . 19

3 Invariance **21**
- 3.1 Introduction . . . 21
  - 3.1.1 Viewpoint variations . . . 21
  - 3.1.2 Illumination variations . . . 22
  - 3.1.3 The need for invariance . . . 23
- 3.2 Geometric invariance . . . 23
  - 3.2.1 Geometric transformations involved in image formation 25
  - 3.2.2 Geometric transformations between two images . . . 28
  - 3.2.3 Geometric invariants . . . 30
- 3.3 Photometric invariance . . . 34
  - 3.3.1 Photometric transformations involved in image formation 34
  - 3.3.2 Photometric transformations between two images . . . 35
  - 3.3.3 Photometric invariants . . . 36
- 3.4 Local, affinely invariant features . . . 36
- 3.5 Conclusion . . . 37

4 Extracting local, affinely invariant regions **41**
4.1 Introduction . . . 41
4.2 Selection of anchor points . . . 43
4.2.1 Harris corner points . . . 43
4.2.2 Local intensity extrema . . . 46
4.2.3 Comparison . . . 46
4.3 Geometry-based region extraction . . . 48
4.3.1 Case 1: Curved edges . . . 48
4.3.2 Case 2: Straight edges . . . 54
4.4 Intensity-based region extraction . . . 59
4.5 Discussion . . . 64
4.6 Conclusion . . . 68

5 Matching local, affinely invariant regions **71**
5.1 Invariant region description . . . 71
5.2 Comparing two regions . . . 76
5.2.1 Comparing two feature vectors . . . 77
5.2.2 Crosscorrelation-based comparison of regions . . . 79
5.2.3 Comparison based on maximum likelihood measures . . 79
5.3 Comparison of the different feature vectors . . . 81
5.4 Matching regions between two images . . . 83
5.5 Semi-local constraints . . . 86
5.6 Conclusions . . . 88

6 Experimental validation **91**
6.1 Introduction . . . 91
6.2 Experimental validation . . . 92
6.2.1 Robustness to viewpoint changes . . . 92
6.2.2 Robustness to image blur . . . 95
6.2.3 Robustness to noise . . . 97
6.2.4 Robustness to illumination changes . . . 98
6.3 Conclusion . . . 98

7 Wide baseline stereo **101**
7.1 Introduction . . . 101
7.2 Experimental results . . . 103
7.3 Conclusion . . . 107

8 Object recognition and localization **109**
8.1 Introduction . . . 109
8.2 Object recognition using local invariant regions . . . 110
8.3 Experimental results . . . 111
8.4 Conclusion . . . 113

9 Content-based image retrieval **115**
9.1 Introduction 115
9.2 CBIR based on local, affinely invariant regions 117
9.3 Experimental results 117
9.4 An application: a virtual tour guide 120
9.5 Conclusion 126

10 Mobile robot navigation **127**
10.1 Introduction 127
10.2 Visual homing 128
10.3 Vision-based mobile robot navigation 135
10.4 Conclusion 138

11 Conclusion **139**
11.1 Summary 139
11.2 Discussion and further research 140

A Affine invariance of function extrema **155**

B Derivation of a geometric semi-local constraint **161**

C Overview of the parameters **165**

D Overview of our database **167**

Nederlandse samenvatting **171**

Curriculum vitae **193**

List of publications **195**

List of Figures

3.1 The effect of viewpoint changes 22
3.2 The effect of illumination changes 24
3.3 Perspective projection 25
3.4 Orthographic projection 27
3.5 The epipolar geometry 30
3.6 Reflection models 35
3.7 Affinely invariant image regions 38

4.1 Example of Harris corner points 45
4.2 Example of local intensity extrema 47
4.3 Construction of geometry-based regions (curved edges) 49
4.4 Geometrical interpretation of $f_2(\Omega)$ and $f_3(\Omega)$ 50
4.5 Example of geometry-based invariant regions (curved edges) . . 52
4.6 Example of geometry-based invariant regions (curved edges) . . 53
4.7 Example of geometry-based invariant regions (function evaluation) 53
4.8 Construction of geometry-based regions (straight edges) 54
4.9 Deskewing based on straight edges 55
4.10 Example of geometry-based invariant regions (straight edges) . 56
4.11 Example of geometry-based invariant regions (straight edges) . 57
4.12 Example of geometry-based invariant regions (preprocessing steps) 57
4.13 Example of geometry-based invariant regions (function evaluation) 58
4.14 Example of geometry-based invariant regions (normalized regions) 58
4.15 Construction of intensity-based regions 60
4.16 Example of intensity-based invariant regions 62
4.17 Example of intensity-based invariant regions (function evaluation) 62
4.18 Example of intensity-based invariant regions (robustness) . . . 63
4.19 Effect of noise, discretization errors and non-linear intensity changes 66
4.20 Alternative construction method 1 67
4.21 Alternative construction method 2 68

5.1 Fischer analysis for generic feature vector 81
5.2 Fischer analysis for normalized feature vector 1 82
5.3 Fischer analysis for normalized feature vector 2 82
5.4 Example of matching regions between two images 85

5.5 Example of semi-local constraints 89

6.1 Robustness to changes in viewpoint, drink can 93
6.2 Robustness to changes in viewpoint, car 94
6.3 Robustness to scale changes 95
6.4 Robustness to image blur 96
6.5 Robustness to image noise 97
6.6 Robustness to illumination changes 99

7.1 Wide baseline stereo, example 1 103
7.2 Wide baseline stereo, example 1 104
7.3 Wide baseline stereo, example 2 105
7.4 Wide baseline stereo, example 2 106
7.5 Wide baseline stereo, example 3 107
7.6 Wide baseline stereo, example 3 108

8.1 Object localization setup 110
8.2 Model images of a coke can 112
8.3 Example of object localization 112
8.4 Example of object localization, matching regions 113

9.1 Examples of successful retrieval 118
9.2 Examples where the retrieval failed 119
9.3 Database retrieval results 120
9.4 Overview of the virtual tour guide system 121
9.5 An overview of the paintings database 122
9.6 A virtual tour guide example, regions 123
9.7 A virtual tour guide example, image retrieval 124
9.8 A virtual tour guide example, image retrieval 125

10.1 Epipolar geometry 131
10.2 Decomposition of the essential matrix 133
10.3 Our mobile robot 134
10.4 Visual homing examples 135

A.1 Photometric invariance $f_2(\Omega)$ and $f_3(\Omega)$ 158

1 Geometrie-gebaseerde extractiemethode voor gekromde randen 178
2 Geometrie-gebaseerde extractiemethode voor rechte randen . . 180
3 Voorbeeld van geometrie-gebaseerde omgevingen 181
4 Intensiteitsgebaseerde extractiemethode 182
5 Voorbeeld van intensiteits-gebaseerde omgevingen 182

Chapter 1

Introduction

To see or not to see, that is the question.
`after W. Shakespeare`

Day after day, we recognize objects, faces and places in just fractions of a second. Moreover, we do so without much effort, more or less in an intuitive way. We do not need to be especially concentrated for it, nor do we have to think hard. Rather, we do it automatically, and we simply take this for granted. Even when we close one eye, we do not loose the ability to recognize things. This means that we do not necessarily need depth information for it. For the same reason, we can also readily recognize the content of a photograph or picture.

We are so familiar with seeing and interpreting what we are looking at that we do not realize its complexity anymore. This makes it hard to understand why the same skills are so difficult to implement on machines. Nevertheless, although object recognition has been one of the prime goals since the very first days of computer vision about fourty years ago, a generic solution has not yet been found.

The difficulty is that, for a computer, a digital image is just a bunch of pixels, an array of numbers between 0 and 255, without any further meaning. To make things even worse, no two arrays representing images of the same object or scene are alike. The position of the object within the image can change, as can its pose, its scale, its orientation relative to the camera or even its apparent color or intensity, due to changes in the illumination.

All these considerations turn computer vision from a boring programming task into the intriguing challenge of making machines see, i.e. making them understand what they are looking at. In this work, we would like to add our own small contribution to this fascinating field.

1.1 Finding correspondences between images

The main contribution of this work concerns what is known as the *correspondence problem*. If one has two images of the same object or scene, given a feature in one image, where is the corresponding feature (i.e. the projection of the same 3D feature) in the other image ? Although humans are able to solve this problem without much effort, it is again not a straightforward task for a computer.

Often, some simplifying assumptions are made to reduce the complexity of this problem. For instance, in most 3D reconstruction algorithms, it is assumed that the images are not too different (i.e. same illumination conditions and similar but not identical camera poses). In that case, the coordinates of the features and the intensity distribution in a small neighborhood around each feature are similar in both images. This in turn allows to restrict the search range and to match features through intensity crosscorrelation.

Alternatively, the complexity of the problem can be reduced by working under calibrated conditions. Indeed, if the internal and external camera parameters are known, the epipolar geometry linking the two images can be derived. In that case, the search range for a corresponding feature can again be restricted. For instance, if we consider point features, the corresponding point is known to lie on the epipolar line. However, in general, the camera parameters are unknown, and deriving the epipolar geometry already requires a minimum number of correspondences, which brings us in a chicken-and-egg situation.

In this work, a novel algorithm is proposed for searching correspondences between images in the most general case – i.e. under wide baseline conditions and without any a priori information about the internal or external camera parameters nor about the scene content.

1.2 Recognizing objects from a limited number of reference views

The inspiration and motivation for starting this work stems from the intriguing field of object recognition.

In contrast to the more classic model-based approaches, *image-based* or *appearance-based approaches* to object recognition have become increasingly popular. Instead of representing an object by some abstract, often three-dimensional model, they simply model an object by a set of images thereof. In this way, they directly exploit what *can* actually be seen in the image instead of what the system developer *hopes* that will be seen, based on the 3D model and some theoretic assumptions on the image formation process.

There are several advantages to this approach. First of all, it can deal with a wide variety of shapes. Recognition has been demonstrated for complex objects that would be very difficult to handle with model-based techniques. Also, it

copes with several types of variations in the same way. Not only different viewpoints are taken into account, the same approach can also deal e.g. with changing illumination. Finally, training the system only requires taking images. There is no need for a three-dimensional, abstract model of the objects.

There are also important disadvantages, however. Building the image database can be a rather painstaking process. The more degrees of freedom there are, the larger becomes the number of images that need to be taken. The method does not extrapolate well, in the sense that recognition needs a model image that is close to it. Hence, model images should densely sample the space of all possible views. Moreover, the object often needs to be separated from the background. This is not an easy task to automate as segmentation is not robust unless objects are e.g. lying on a homogeneous background. Using a simple bounding box is not necessarily enough if backgrounds change a lot.

Schmid and Mohr [SM97] proposed an elegant solution to alleviate these drawbacks. They extract local, invariant features from the image and match these with the features extracted from the model images. The locality of the features suppresses the need for segmentation and makes the system robust to partial occlusions and image clutter. The invariance allows to generalize from a single image and hence reduces the number of reference views needed. However, invariance in their case is under 2D rotation and translation only.

In this work, we increase the level of invariance from pure geometric to geometric and photometric, and from 2D rotation and translation to full plane affine transformations. In this way, changing the viewpoint will have no effect on the computed invariants (as long as the same part of the object is still visible and some basic assumptions are fulfilled). As a result, one reference image can cover a wide range of viewpoints. For a two-dimensional object, a single reference image suffices to model the object, while three-dimensional objects can most often be represented by less than ten or twenty images (depending on the complexity of the object).

Achieving this level of invariance on a *local* scale is not a straightforward extension though. It implies the need for local, affinely invariant regions, as will be explained in chapter 3. How to extract such local invariant regions has been the main topic of this work.

1.3 Other applications

Object recognition is not the only application of wide baseline stereo matching. Finding correspondences between wide baseline images is a basic tool that can be used each time one needs to link several images. As a result, applications can be found in many different sub-domains of computer vision.

3D reconstruction The accuracy of a 3D reconstruction increases with increasing baseline. However, due to the lack of robust matching algorithms under wide baseline conditions, most 3D reconstruction algorithms still work

with small baselines only, possibly using a sequence of images in combination with feature trackers. Apart from accuracy, also the flexibility of such systems could benefit from a wide baseline matching algorithm.

Content-based image retrieval Due to the tremendous increase in available digital information, there is an increasing need for algorithms to automatically search through large multimedia libraries, not only based on manually added text annotations but even more so directly based on the image or video content. Most attention in this relatively new field of research has gone to finding *similar* images, looking at rather global color and texture features. In addition to this type of queries, finding other occurrences of the *same* object or scene also represent important queries, typically requiring more local and structure oriented features. The affinely invariant regions proposed in this work seem well suited for this task. They can to a large degree generalize from a single image, both with respect to viewpoint as to illumination conditions, and they are robust to occlusions and image clutter. Maybe the biggest advantage is that they allow the use of *indexing* or *hashing* techniques, making the retrieval time less dependent on the size of the database. Content-based image retrieval is the main topic of chapter 9.

Mobile robots More and more mobile robots nowadays are equipped with one or more cameras. Most object recognition algorithms can be applied in this context for *place recognition* or *robot localization* (e.g. after a power shut down), or simply to localize moving objects the robot should interact with.

Also, some basic information about the robot's relative position can be derived from correspondences between the currently seen image and a reference image taken from a known location (assuming a static scene). In chapter 10, we describe how this can be used as part of a *visual homing* experiment, where the robot autonomously moves itself to a fixed position that has been specified by a target image only. Moreover, two possible strategies are described to develop a fully vision-based navigation system based on the techniques developed in this work.

Image mosaicing If the scene one is looking at is planar or the camera is undergoing a pure rotation, several images can be glued together to form one large picture or *mosaic*. This allows to obtain an image with higher resolution or larger dimensions than would be possible using a single image. Alternatively, mosaicing can be used as a compression technique for video sequences, supplying one large image for the background once instead of sending it over for each shot again and again. Finally, image mosaics are often used in web-presentations and virtual reality applications.

However, gluing images together is only possible if one knows the underlying transformation. This can easily be derived once at least four initial correspondences are available. Our wide-baseline stereo matching techniques can be used

to find these correspondences automatically.

Grouping *Grouping* is an important step in vision that combines bits and pieces of visual information within an image into higher-order perceptually salient structures, more amenable to semantic interpretation. As such, it is an important stepping stone between low-level vision and scene understanding, leading towards a deeper understanding of observed shapes structure and scene organization.

By looking for similar, affinely invariant regions within an image, repeated, planar (but not necessarily coplanar) patterns can be detected. These can then be used to build a hypothesis about the specific regularity in the image.

1.4 Main contributions

From an abstract point of view, this thesis aims to push the limits of the appearance based approach to object recognition by introducing more invariance on a local scale. This allows to reduce the number of reference views and hence increases the efficiency and flexibility.

Before we enter a more detailed discussion of these topics, it seems useful to summarize the main contributions we believe have been realized in this work:

- Two methods for *affinely invariant region extraction* have been proposed. This allows to compute photometric invariants on a local scale under more general transformations than simple 2D rotation and translation. Although it has been developed independently, it can be considered a generalization of the automatic scale detection developed by Lindeberg [Lin93, Lin98]. Based on these local affinely invariant regions, the feasibility of an affinely invariant photometric description on a local scale has been demonstrated. This work was published in [TuyAl99a, TVG99, TVG00, TTMVG01, VGTT01].

- The affinely invariant region extraction has been integrated in a system to automatically match regions between different views of the same object. Each region is described by a feature vector of affinely invariant, generalized color moments, allowing an efficient implementation of the matching through the use of indexing techniques. Furthermore, false matches caused by symmetries in the image or lack of discriminative power of a local region are rejected based on two simple semi-local constraints. Due to the affine invariance, correspondences can be found even in case of large changes in viewpoint [TuyAl99b, TVG00].

- The combination of affinely invariant region extraction and matching has been used in different application domains. First of all, it is useful for object recognition purposes, with less reference views than needed for the more traditional appearance-based approaches [TuyAl00a]. Another

application domain is content-based image retrieval [VGTP99, TVG99, VGTT01]. Here, the goal is to retrieve similar images from a whole database of images. Finally, the local invariant features have also been used in the context of mobile robot navigation, where an algorithm for visual homing has been developed [TuyAl99a].

1.5 Outline of the thesis

Here, we briefly sketch the outline of the thesis and the contents of each of the individual chapters.

Chapter 2 gives a short overview of the state of the art, with the main focus on object recognition. It is by far not our intention to give a complete overview of the vast literature available on this topic. Rather, we select a few works, that have been most influential for our own work or that are in one way or another related to it.

We continue in *chapter 3* with a short recapitulation of the use of invariance in computer vision. Special attention is given to the derivation of invariants on a local scale. Here, it becomes clear that obtaining more general invariance than simple 2D rotations and translations on a local scale is only possible if we can in one way or another define the same support in the different views. To this end, we introduce the concept of local affinely invariant regions.

Chapter 4 discusses how we extract such local affinely invariant regions from an image. Instead of trying to find one universally applicable type of regions, we follow a more heuristic approach. Our goal is to come up with an opportunistic system, that exploits several region types, depending on what's on offer in a specific image. The first type of regions we propose are coined geometry-based regions, as they firmly rely on the presence and accurate detection of some geometric entities. The second region type, on the other hand, directly starts from the raw intensity data, so we refer to it as the intensity-based region extraction method.

Chapter 5 deals with the invariant description of the regions using a feature vector of moment invariants. Several feature vectors are proposed, as well as different methods to compare two regions. Next, we have a closer look at the actual matching process, i.e. given the regions in two views of the same object or scene, how to find the corresponding ones. Also, some semi-local constraints are proposed, that can be used to filter out the false matches and hence increase the robustness of the system.

Before we look at some specific applications, we first perform some experiments to validate the robustness of the proposed matching algorithm to different sources of image variability, such as viewpoint changes, illumination changes, noise and image blur in *Chapter 6*.

Then, we focus on some possible applications of our work. *Chapter 7* deals with wide baseline stereo, i.e. deriving the epipolar geometry linking two images of the same object or scene. Next, *Chapter 8* discusses some experiments on

object recognition and object localization. We continue in *Chapter 9* with the application of the local invariant regions in the context of content-based image retrieval. *Chapter 10* discusses an algorithm for visual homing of a mobile robot, based on local invariant regions, and some possible extensions to vision-based robot navigation.

Finally, the conclusions of our work are presented in *Chapter 11*. To enhance the readability of this text some more tedious mathematical derivations were placed in appendices.

Chapter 2

State of the art

Everything that can be invented has been invented.
C. H. Duell, 1899.
US Office of Patents

In this chapter a short overview of the state of the art is given. We start in the domain of object recognition (section 2.1), as our technique can be seen as an integration of the model-based and appearance-based approaches in this context. Special attention is given to methods using local features in one way or another. Next, we focus on different ways to overcome the large image deformations that occur even on a local scale, due to changing viewpoints. Finally, in section 2.2, we draw our conclusions. For a more detailed overview of the literature specific to the different applications (wide baseline stereo, content-based image retrieval and vision-based robot navigation), we refer to the corresponding chapters.

2.1 Related work

Object recognition is a subject that has been studied in computer vision since its earliest days (about fourty years ago). Most systems in those days worked rather *ad hoc* on simplified objects such as polygons and polyhedrons. Here, we'll focus only on the more recent and more fundamental approaches to object recognition, paying special attention to those techniques that are most related to our work. For a more complete overview of the object recognition literature, we refer to [Pop94].

2.1.1 Model-based versus appearance-based object recognition

In general, two different schools can be distinguished: the model-based approaches and the image- or appearance-based approaches to object recogni-

tion. [1]

The classical, model-based approaches to object recognition start from an explicit 3D model of an object's shape. 2D features or primitives such as lines, holes, circular segments, etc. are extracted from the image and matched to the 3D model of the object, taking into account the projection from 3D to 2D (e.g. [Low87]). Alternatively, 3D volumetric primitives can be extracted directly from the image to be matched with the various models (e.g. [DP92]). Often, graphs are used to represent the spatial relationships between features. Recognition in those cases is a question of (sub)graph matching. The main drawback of the model-based approaches is that they firmly rely on the feature extraction, which is often a vulnerable, error-prone step. Also, this approach is only feasible for special object classes, composed of primitives that allow easy and robust detection.

In response to the great difficulties encountered in the extraction of inherent characteristics in the model-based approaches, some remarkably successful alternatives have been proposed, belonging to the school of appearance-based approaches. Instead of starting with some difficult and often inaccurate feature extraction, these techniques are based on raw image data, so they require little image processing overhead. They also tend to require less or no camera, robot, and/or hand-eye calibration. The model of an object or scene is extracted directly from its images. These approaches can usually deal with complicated objects and scenes because they do not have to analyze the data. For the very same reason, they are more vulnerable to changing backgrounds and scene clutter, however. Another disadvantage is the extensive learning phase. Many images have to be taken, both for training (as allowable scene variability should be learned), as for operating the system, because reference images should densely sample the space of possible views. This is a definite disadvantage, that implies that such techniques can only be used in controled worlds, with not too many objects and not too many changes in the viewing conditions. Nevertheless, there is some evidence that the human visual system uses an appearance-based representation for object recognition [Ull89], as humans are able to recognize objects more accurately and rapidly when they are seen from particular viewpoints, implying that those views of an object are readily available while others must be computed as needed.

While the model-based approaches can be considered as being more object-centered, the appearance-based approaches can be thought of as being more viewer-oriented. Another difference is that, in general, the appearance-based techniques exploit to a far greater extent the available photometric information (sometimes even exclusively, such as the systems based on color histograms [MN95]) while the model-based approaches focus on the geometric entities present in the image. As a last distinction, one can point out that the appearance-based approaches are traditionally more empirical, while the model-based approaches try to analytically model the relation between 3D ob-

[1] It should be emphasized though that this classification is not complete and by no means univocally adopted.

ject features and their projections in the image.

A good example of a model-based approach to object recognition is the geon-based recognition system developed by Biederman [Bie92]. He has proposed a theory of real-time human object recognition that posits that objects and scenes are represented as an arrangement of simple volumetric primitives, termed *geons*, characterized by specific, viewpoint-invariant properties such as whether a given contour is straight or curved, or whether pairs of such contours are parallel or not. The edges detected in an image are first combined into geons, which are in turn matched to the object models in order to recognize the objects.

Traditionally, object recognition systems using *invariance* [MZ92, ZisAl94] also follow the model-based philosophy. This strongly mathematical approach extracts features that remain unaltered under specific transformations. Recognition times are usually short thanks to so-called 'indexing' or 'hashing' techniques. Most attention has gone to invariance under geometric transformations, but also changes in illumination can be dealt with using invariance (e.g. [NB96, NG98]). The problem is that no invariants exist for general 3D objects under general viewing conditions starting from a single image [BWR90, CJ91].

Recognition based on color histograms, as first introduced by Swain and Ballard [SB91], is an example of an appearance-based approach. They were among the firsts to recognize that color could be used as a direct clue to recognize objects and images. In their system, objects are recognized by computing a color histogram, which is then compared with the color histogram of each model image using a *histogram intersection technique*. Most applications of this approach can be found in the context of content-based image retrieval.

The SLAM system developed by Murase and Nayar [MN95] (Software Library for Appearance Matching) is another example of the appearance-based school, which has been especially influential. Their system can be considered an extension of plain template matching. Image data are correlated against principal components of a set of training images (so-called *eigenimages*). Different objects are represented by different manifolds in the space spanned by a subset of the principal components. Although no on-line feature extraction is applied, the method can efficiently recognize hundreds of objects from a range of viewpoints.

Also worth mentioning in this context is the alignment technique developed by Ullman [Ull89, UB91]. With three views of a rigid object whose contours are defined by surface tangent discontinuities, they showed that it is possible to compose a fourth view simply by interpolating among these three views with a linear operation. If the object has smooth contours, the appearance of its rim is somewhat harder to predict. Nevertheless, six views yield a good interpolation even in this case. This result is useful as a final verification step in any appearance-based approach.

Finally, systems based on *aspect graphs* [KVD79, BahAl97] can be considered as a hybrid technique. As with appearance-based methods the idea is to store several views of an object, rather than a 3D model. However, the refer-

ence views are carefully selected based on the topological equivalence between views, which requires quite some feature extraction and reasoning about the model. On the viewing sphere, representing all possible viewing directions, areas of topologically equivalent projections are deligned. Usually the information is stored in a graph. The aspects are the nodes and the arcs between them indicate the crossing of a cell boundary between two neighboring cells. However, the complexity fastly grows beyond acceptable limits when working with real objects. Moreover, the topological changes are not always detectable in the image.

2.1.2 Object recognition techniques based on matching local features

As an attempt to combine the advantages of the model-based and appearance-based approaches into a single system, methods based on the matching of local features have recently been proposed. As in the appearance-based approach, an object is modeled based on a set of images thereof, taken from different viewpoints and under different illumination conditions. However, as in the model-based approaches a fair amount of feature extraction is added, both in the learning phase as in the recognition phase. The object is no longer represented by the raw image data, but rather by a set of local features, extracted from the reference images. During the recognition phase, the local features extracted from the image are compared with the features in the database of reference images to select the reference image with the most similar features and hence to recognize the object.

The local character of the features makes them immune to occlusions. Indeed, if part of the object or scene is invisible because it is occluded by another object, only the local features on this part of the object are lost, while the other features 'survive' the occlusion. As long as the occlusion does not cover the whole object, the remaining features can still be used to give a (partial) description of the object. The same holds in the case of self-occlusion.

At the same time, the use of local features allows to deal with image clutter as well. These are other objects present in the image that may distract the attention from the real object of interest. Only part of the extracted local features originate from the object of interest in that case, while the others stem from the image clutter. However, this need not be a problem, as not all local features must be mapped between both images before one can draw conclusions. As a result, no background segmentation of the image is needed as a preprocessing step, in contrast to the traditional appearance-based approaches working on the entire image.

The use of local features also has some disadvantages though. First of all, as they only cover small portions of the object, they force the system to look at details instead of looking at the global appearance or trend of an object. As a result, they are less suited to generalize from identical objects to similar objects (i.e. perform object classification rather than object recognition). Also,

different objects made of the same components will be hard to distinguish, as they share the same local features.

Lamdan and Wolfson [LW88a] were the first to transform object recognition into a point-matching task. Their method, which they coined *geometric hashing*, finds correspondences between two images based on the geometric configuration of local features. More precisely, they select three points that serve as an affine basis of a coordinate system and express all other points as coordinates in this basis. This allows to recognize objects even if they are only partially visible. Moreover, the method can be made quite efficient through the use of indexing techniques.

Later on, their approach was improved by Rigoutsos and Hummel [RH93], who reduced the noise sensitivity through a probabilistic voting scheme.

A drawback of this method, however, is that it is based on combinations of several points, exploiting their geometric configuration. As a result, it does not really work on a local scale. Moreover, the affine measure they use is *not* viewpoint invariant, unless the selected points are coplanar. Also, it was soon found out that using only point features may result in a large number of false positives, as the individual points cannot be distinguished. Therefore, also other features were tried out [LW88b, Wol90]. This also explains why most later work focuses on the matching of individual features which can be discriminated based on a local description of their immediate neighborhood.

Ohba and Ikeuchi [OI96, OI97] for instance apply a direct extension of Murase and Nayar's eigenimages to local features, using so-called *eigenwindows*. An object is modeled by multiple, partial appearances in an eigenspace. To this end, each image is divided in smaller windows. Good window candidates are selected based on detectability, uniqueness and reliability.

A similar approach has been adopted by Colin de Verdière and Crowley [CVC98], who combine the local descriptors in a *local appearance grid*, while Krumm [Kru97] has proposed to use local, binary images instead to better withstand photometric changes due to changes in the illumination.

A main disadvantage of the eigenimage-approaches however is the difficulty of extracting good statistics and the data dependency of the eigenimages. Each possible source of variation (be it illumination, pose or orientation), should be learned by the system through examples. Moreover, the example images should densely sample the space of possible views. This extensive learning phase can be circumvented by using a fixed set of orthogonal basis functions instead, which are suboptimal but data-independent. A Fourier transform, for instance, can be interpreted as a decomposition into orthogonal complex trigonometric patterns. Applied on a local scale, this results in a bank of Gabor filters, which are often used to discriminate textures. Other examples include the discrete cosine transform, the Hadamard transform or the Haar transform. Projection onto derivatives of Gaussians has proven especially successful in this context, as it has been shown that the eigenimages of a large set of image patches resemble derivatives of Gaussians [HanAl91, RB95]. Moreover, studies on the

biological visual system suggest that the receptive fields in cortex visual cells can be modeled as derivatives of Gaussians. Another advantage is that one can exploit the *steerability* of these functions [FA91]: it is possible to calculate the nth order derivative of an arbitrary orientation ϕ as a linear combination of a finite number of nth order derivatives. This has been used, among others, by Ballard *et al.* [BW93, RB95], who use a feature vector of nine oriented Gaussian filters at five different scales.

Instead of 'normalizing' each new measurement using the steerability of the Gaussian derivatives, several derivatives can also be combined into expressions that are *invariant* under 2D rotation. An excellent example of this approach is the work of Schmid and Mohr [SM97], who use an invariant feature vector as derived by Koenderink and Van Doorn [KVD87] to distinguish the local features. Another remarkable point is that they focus on a set of *interest points* only, instead of applying the filters to the entire image. This work served as a departure point for our work, so we will discuss it in more detail in a separate paragraph (see section 2.1.5).

Others have tried to include color information in the local feature descriptor to increase the selectivity of the local features. The idea is that in order to distinguish features on a local scale, one should use all the information at one's disposal. Using color allows to discriminate features that would otherwise stay indistinguishable. Ennesser and Medioni [EM95] therefore use local color histograms, which, thanks to the histogram-based approach, are rather robust to geometric transformations. However, similar to the original (global) color indexing proposed by Swain and Ballard [SB91], this approach is sensitive to illumination changes. Indeed, changes in the spectrum or intensity of the light sources, or even changes in their position relative to the object and/or camera, can result in completely different color values in the image. Slater and Healey [SH96] therefore add illumination invariance by exploiting *normalized* local color histograms, exploiting the affine property of color distribution change.

Nayar and Bolle [NB96] on the other hand propose to use reflectance ratios to overcome the changes in illumination. They match regions with consistent colors given the partitioned model and image derived by some other color properties. A disadvantage of their system is that it requires a preliminary segmentation of the image into regions of constant reflectance. Nagao and Grimson [NG98] solve this problem by combining photometric invariants with geometric constraints. More precisely, they look at reflectance ratios over edges.

Finally, we refer to the work of Startchik, Milanese and Pun [SMP98], who combine geometric and photometric information in a *projectively* invariant description. They start with the geometric construction in an invariant way of three convergent lines, that form a projective basis for all lines emanating from the common intersection point. Then, the intensity patterns along rays are used for the final description. However, their method relies on up to four disjoint curves, so it is not very local any more.

Of course, the overview given in this paragraph is far from being complete. Many others have used local image descriptors in one way or another. An example of a totally different approach is the work of Viola [Vio96], who uses some learned complex local features.

2.1.3 Object recognition techniques based on statistics of local features

In the previous paragraph, an overview of object recognition systems based on the matching (or recognition) of individual local features has been given. As an alternative, some systems have been developed that describe an image or object using local features, but *without* trying to find individual feature correspondences. Instead, they focus on the overall statistics of the local features within the image.

The best known example of this approach is the recognition system based on color histograms developed by Swain and Ballard [SB91], which we already discussed in paragraph 2.1.1. Indeed, color can be considered as a local feature as well, as it is a characteristic of a single pixel, the most local feature of all. Albeit not very selective on its own, the global statistics of such feature can be used to discriminate different objects.

The main drawback of this method however is again its sensitivity to changes in the lighting conditions. Hence, several variations on the original color indexing technique as developed by Swain and Ballard have been proposed to include more illumination invariance. Healey and Slater [HS94] for instance calculate moment invariants of the entire color histogram to compensate for a constant intensity change over the entire image. Funt and Finlayson [FF95] have proposed 'color constant color indexing'. They use derivatives of logarithms of the color channels in order to provide illumination invariance under the assumption of a locally constant illumination.

Still, many object classes cannot be described or recognized by color alone. That is why color histograms have been combined with geometric information in a variety of ways [HuaAl99, MMK95, Mel97, PZ96, SH95, SB98]. In particular, the SEEMORE-system of Mel [Mel97] uses more than a hundred different feature channels, including color, intensity, cornerness, contour shape as well as texture features derived using Gabor filters. Pass and Zabih [PZ96] use a *Color Coherence Vector*, which partitions each histogram bucket based on spatial coherence. Huang *et al.* [HuaAl99] propose *Color Correlograms* instead of color histograms, which express how the spatial correlation of color changes with distance.

Finally, Schiele and Crowley [SC96] do not use color at all, but multidimensional receptive field histograms, i.e. histograms based on vectors of local neighborhood operators composed from Gaussian derivatives.

2.1.4 Increasing the robustness to viewpoint changes

In the previous paragraphs, we already gave some examples of how researchers have tried to make the local feature descriptors or statistics more robust to illumination changes. Far less has been done to increase further the robustness to *geometric* deformations though (except maybe for the development of robust histogram intersection techniques). Geometric invariance is in most cases limited to 2D rotation and translation.

The problem arises from the fact that no neighborhood with fixed shape and size exists that remains unaltered under more general transformations. A circular neighborhood for instance, which perfectly survives 2D rotations and translations, is transformed to a bigger or smaller circular neighborhood under a scale change, and to an elliptic neighborhood under an affine transformation. As a result, it is impossible to obtain invariant descriptors under this type of transformations based on local neighborhoods of fixed shape and size. This is a serious problem, as *any* image measurement (moments, derivatives, etc.) is implicitly or explicitly computed over a certain neighborhood.

With respect to the changes in scale, this problem is often coped with by applying a *scale space approach*, i.e. the same features are computed several times with different support (i.e. over several neighborhoods with different size). Schmid and Mohr [SM97], for instance, compute the same feature vectors for different scales, each time using circular neighborhoods taken a scale factor of 1.2 apart from each other. For more general transformations, these problems are circumvented by densely sampling over all possible viewpoints.

However, it is clear that these are not very elegant solutions, both with respect to the memory requirements or efficiency as with respect to flexibility and ease of modeling of a new object. With more degrees of freedom (e.g. when considering affine transformations), it becomes completely unfeasible. Nevertheless, an affine scale space has been applied by Slater and Healey [SH96], who computed for each model image normalized local color histograms for several elliptic neighborhoods corresponding to different slant and tilt angles. However, the number of objects and neighborhoods in their experiment was very limited.

Alternative methods are needed, which automatically select the correct neighborhood in an invariant way, i.e. such that the neighborhoods cover the same physical part of the object, adapting their shape or size based on the local color or intensity profile as to compensate for the geometric transformations.

Lindeberg [Lin93, Lin98] has proposed a method coined *automatic scale detection*. He considers circular neighborhoods with varying radius (i.e. linear scale space), and looks for maxima over this scale space of combinations of second order normalized derivatives such as the Laplacian to select an appropriate scale. For different features (corners, edges, ridges, etc.) different normalized differential entities are proposed.

This automatic scale detection has first been applied for object recognition purposes by Colin de Verdière and Crowley [CCVC99, HCVC00]. They do

not only apply automatic scale detection, but also automatic orientation detection. This allows to normalize a vector of receptive fields accordingly before projecting the local neighborhood onto this vector.

Lowe [Low99] on the other hand integrates the automatic scale detection with the interest point detection. He uses maxima over scale space of a difference-of-Gaussians filter. Special attention has been paid to efficiency in their system.

The examples given above are still limited to changes in scale (on top of the 2D rotation and translation). In this work, our goal is to extend this idea to the automatic selection of *affine* neighborhoods, invariant under affine transformations. From now on, we will refer to this kind of neighborhoods as *affinely invariant regions*. We found very few other work focusing on this subject. Those we found were mostly situated in the context of texture analysis.

Ballester and Gonzales [BG98] have developed an iterative method to find affinely invariant regions in textured images, based on an affinely invariant analogue of the second order moment matrix. For each point in the image, they determine an elliptic neighborhood over which the histogram is stable based on the underlying texture pattern. Implicitly, their method relies on the fact that for images of textures the second order moment matrix remains more or less constant with varying region parameters. For general images however, this assumption may well be violated, such that convergence is no longer guaranteed.

The multi-scale blob-detection method developed by Lindeberg [GL96] in the context of shape from texture is another example where elliptic regions are extracted from an image. Their computations are based on a local covariance matrix $\mu_{\mathbf{L}}$, which represents the local statistics of the first order Gaussian normalized derivatives of image brightness. The computation of this matrix, from which an elliptic neighborhood can be derived, involves two scale parameters to be determined: a local scale describing the amount of smoothing used in the derivative computations and an integration scale defining the region over which the statistics are computed. The determination of these scales follows the automatic scale detection method based on extrema over scales of functions derived from $\mu_{\mathbf{L}}$. The same principle is applied to guide the selection of where in the image to compute the texture descriptors, but now based on second order derivatives in order to favor centers of blob-like structures (e.g. based on the Laplacian).

However, as such, their system is *not* affinely invariant yet, as it uses only a linear scale space. An extension is presented in [GL94], where an affine scale space is used instead, applying non-uniform Gaussian smoothing kernels. The local scale and the integration scale have both been replaced by 2×2 scale matrices, increasing the number of parameters to be determined to six. An iterative procedure for affine scale adaptation is performed that is hoped to converge to the fixed point where the local covariance matrix μ_L is equal to the matrix used for shape adaptation. However, for many images, there are (at least) two more fixed points, one corresponding to the shape adaptation

matrix being singular and one corresponding to an infinite amount of isotropic smoothing. Maybe that explains why the experimental results for the affine scale space method have been limited almost exclusively to synthetic images.

2.1.5 The approach of Schmid and Mohr

Our point of departure is the work by Schmid and Mohr [SM97], who developed a particularly interesting approach. It heavily leans towards using a view-based approach, but at the same time it does a fair amount of feature extraction in that local invariants are extracted. On the one hand, their approach inherits from the view-based school the ability to deal with objects of high complexity. On the other hand, their approach is less sensitive to scene clutter and occlusions and it can to a certain extent generalize from a single view, advantages of using local invariants.

Through the invariance, inherent characteristics that change with viewing direction are filtered out. What is left is an impoverished description of the shape, the more so as the features cover only part of the object's surface. Yet, the locality of the features and the level of invariance in the description render the technique efficient. The invariance has the advantage that the models can be obtained directly from images, while the technique is at the same time more robust than those based directly on raw image data. Occlusions, partial visibility, and scene clutter can be handled due to the locality of the patches. Not all of them need be visible and at least a few probably are not. This technique will also be immune against some changes never actually shown to the system during training. This is a problem when raw image data are used. Indeed, as the degrees of freedom in relative camera motion and changes of illumination increase, it quickly becomes virtually impossible to collect a representative set of reference views. The invariance lowers the dimension of the problem.

The idea is to first identify special points of interest, e.g. corners, and further on concentrate on these points only. Each interest point is described by a nine-dimensional rotation invariant vector of local characteristics based on Gaussian derivatives, as originally proposed by Koenderink and Van Doorn [KVD87]. Scaling is handled by computing the same feature vector over several circular neighborhoods of different sizes. The invariance ensures that the number of reference images needed to recognize an object can be reduced. In particular, shifting or rotating the object in the image will have no effect on the invariants and hence a single reference image in the model suffices to cover all these cases. The vector responses of all interest points are stored in a hash table indexed by the nine-dimensional feature vector. Comparison of feature vectors is based on the Mahalanobis distance.

The principal application of this approach is the correspondence problem between a test image and the reference images stored in the hash table. In addition, the approach is suitable for object (or image) recognition which can be seen as a correspondence problem. Recognition requires that similar points of interest are extracted from the test image, together with their invariant rep-

resentation. These are then used in a voting algorithm, to select the image in the database with the highest number of similar features. The voting technique is made more selective by combining the vector responses with semi-local constraints between different interest points, checking the spatial configuration. This has been shown to significantly reduce the number of false matches. Recently, a probabilistic voting scheme has been presented as a further improvement to the system [Sch99].

2.2 Conclusion

From this overview of the state-of-the-art in object recognition, we draw the following conclusions, which represent some vital choices we made for the development of our own system.

- **Appearance-based approaches** seem to hold good promise, as they start from what can actually be seen in the image; they allow easy modeling of objects (simply by taking images thereof) and they are not limited to specific object classes but, on the contrary, can deal even with very complex objects.
- **Local features** help to overcome the drawbacks of the classic appearance-based approaches: they make the recognition immune to partial occlusions and image clutter and they avoid the need for segmentation prior to recognition.
- **Invariance**, both under geometric and photometric transformations, allows the system to generalize from a single view, and hence to reduce the number of reference images needed to model an object.

In our opinion, the weakest point so far is the level of invariance under geometric transformations. Most systems only include invariance under 2D rotation and translation. To avoid more complex geometric deformations, the viewing sphere is sampled more densely or a scale space approach is applied. Appearance-based systems that can deal with larger geometric and photometric changes are very rare [2]. This results in a high number of reference images needed to represent a single object from an arbitrary viewpoint. To the best of our knowledge, no appearance-based system is yet available that can deal with completely arbitrary viewpoints. Most are limited to rotations around a fixed axis. This can be explained by the fact that gathering this high amount of reference images is quite a painstaking process. In this work, we focus on how we can increase the invariance to arbitrary viewpoint changes while keeping the local character of the features.

In this chapter, we mainly looked at object recognition systems. However, many other applications exist that could benefit from a method allowing to

[2] apart from the color histogram-based techniques, but these are seldomly sufficiently discriminative to distinguish a large number of objects

match local features under affine deformations. Retrieving images of the same object or scene in a library of digital images is a typical example. Here, one typically has only one reference image available, and successful retrieval depends on the ability of the system to generalize over viewpoint changes and lighting conditions. Other typical applications include wide baseline stereo or vision-based mobile robot navigation. For a short overview of the state of the art in each of these sub-domains, we refer to the corresponding chapters.

Finally, we would like to stress once more that the overview given in this chapter is far from being complete. We just selected a few works that seemed most relevant to our work. For a more complete overview of the literature, we refer to [Pop94].

Chapter 3

Invariance

This is not mathematics, it is theology.
P. Gordon[1]

In this chapter, a brief overview of invariance in the context of computer vision is given. After a short introduction explaining the need for invariance (section 3.1), we focus on geometric invariance in section 3.2. First, we discuss the transformations that govern the image formation process, both for planar and three-dimensional objects. Next, the transformations between two images of the same object or scene are derived. This in turn allows to extract geometric invariants. More or less the same scheme is followed for the photometric transformations and invariants in section 3.3. We continue in section 3.4 with a short discussion of invariance on a local scale, explaining the need for affinely invariant regions in this case. Section 3.5 concludes the chapter.

3.1 Introduction

Two different images of the same object or scene are never completely identical. As a consequence, we cannot simply compare the raw image data (e.g. through a pixel-wise comparison) to find out whether the images contain the same object or not. On the contrary, an image is the combined result of many factors, including illumination, viewpoint, sensor type, etc. Changes in these factors can lead to important variations in the image appearance. In this section, two main factors, namely viewpoint and illumination, are described and their impact on the image appearance is illustrated.

3.1.1 Viewpoint variations

In Figure 3.1, we show the effect of changing the viewpoint on a sample image. The relative position and orientation between the camera and the object clearly

[1] On being exposed to Hilbert's work in invariant theory.



Figure 3.1: *Two images of the same object, taken from different viewpoints.*

have a large effect on the appearance of the object in the image. Moreover, for general 3D objects, there is no simple, geometric transformation linking a point in one image to the corresponding point in the other image. This can be explained by the fact that an image is only a two-dimensional projection of the three-dimensional world. The problem arises from the fact that information is lost under projection, which is a non-invertible operation.

The effect can be described as some kind of geometric distortion between the images, with some parts becoming larger while other parts typically get smaller. As a consequence, it is clear that relative distances are not preserved. The same holds for relative orientations or the shape of curves and object contours. As a result, direct measurements in the image cannot be used to specify an object, as these measurements change from view to view.

To make things even worse, also the visibility of the objects is influenced by their relative position to the camera, causing partial occlusions of one object by another as well as self-occlusions.

3.1.2 Illumination variations

Another factor highly influencing the appearance of an image is *illumination*. Both the position and the color or spectrum of the light source(s) to a large degree determine the intensity and color of the viewed objects. Since the human eye and brain are very good at compensating for this effect, it is often –wrongfully– underestimated. Figure 3.2 illustrates its importance. Three images of the same object were taken from the same viewpoint but under different illumination conditions. These are shown on the upper row of Figure 3.2. To each image, we added the same reference line in white. The intensity profiles along these lines for each of the three color-bands (R, G and B) are added below the images (three curves corresponding to the three different images for each colorband). From these different intensity profiles, it becomes clear that the different illumination conditions have a direct influence on the intensities and hence on *all* features extracted from the image. For instance, in the blue band, the value of the same pixel can vary from 100 in one image, over 120 in

another image to 180 in the third image, and similar variations can be found in the other bands.

3.1.3 The need for invariance

From the examples given above it is clear that changes in viewpoint or illumination, combined with other effects such as image noise, changing backgrounds, use of different cameras and so on can result in large variations in the image appearance. Nevertheless, a good descriptor of the image should be immune to all these changes, yielding similar descriptions under all circumstances. The reliability of such a descriptor depends on how well it can separate the information influenced by viewing conditions from real object properties, and use the latter. Separable object properties are called *invariants*. These are features that remain unaltered in spite of the variations in the image appearance caused by changes in the image acquisition process. Initially, they were developed to allow arbitrary viewpoints, but soon it was realized that, as an extra advantage, the intrinsic camera parameters need not be known either. For an overview of invariants and their application to computer vision, we refer to [MZ92, MZ94, Woo96].

Note that, as an alternative approach one could try and eliminate all sources of variations, e.g. by fixing the camera position and carefully controlling the lighting conditions. Assumptions about the characteristics of the acquisition environment can then be used to simplify the feature extraction process. Such controlled environments are typical for state-of-the-art machine vision systems in industry. However, for many applications controlling the acquisition parameters is not an option. In object recognition, the goal is exactly to recognize an object in spite of all this variation. Wide baseline stereo is another example. Compared to small baseline stereo it has the advantages of more flexibility and a higher accuracy in the resulting 3D reconstructions at the extra cost of having to deal with the larger image variations (e.g. during the search for correspondences).

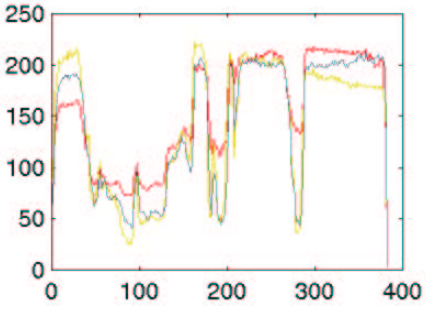
A last alternative is to increase the number of reference views, storing images for all possible variation, but, as already pointed out in the previous chapter, this is not desirable either.

3.2 Geometric invariance

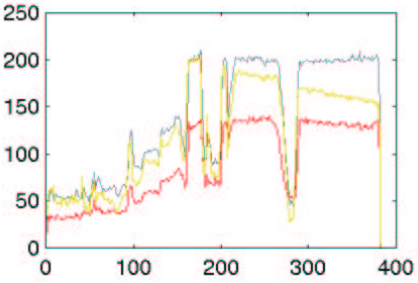
Invariants are mathematical expressions that combine different image measurements (intensities, image coordinates or even derivatives) in such a way that the result does not change under a specific group of transformations. So the first problem in finding invariants is the identification of the transformation group that underlies the problem at hand. That is why we start by investigating the transformations that govern the image formation process (section 3.2.1), followed by the transformations that link corresponding points in two images



Red band



Green band



Blue band

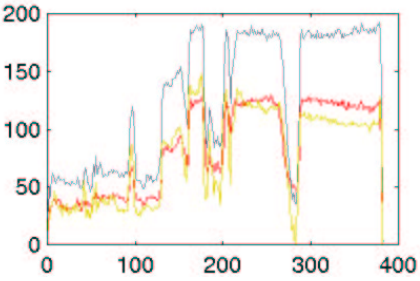


Figure 3.2: *Three images of the same object, taken under different illumination conditions, and the intensity profiles for the red, green and blue band respectively (with the color of the curves more or less similar to the color of the light source in the corresponding image).*

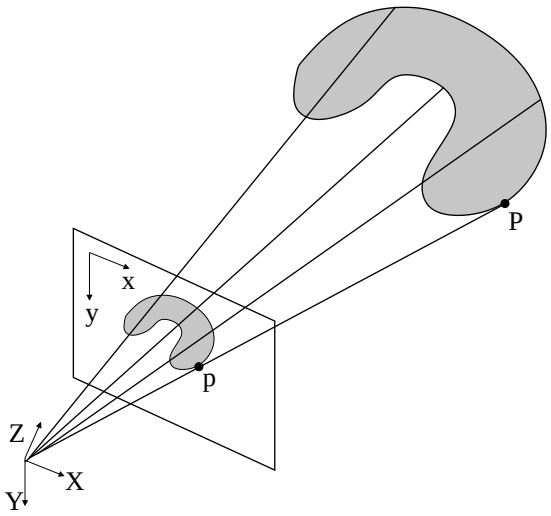


Figure 3.3: *Perspective projection.*

(section 3.2.2). Each time, we make a distinction between planar (2D) objects and general 3D objects. After that, we are finally ready to deal with the geometric invariance itself (section 3.2.3).

3.2.1 Geometric transformations involved in image formation

Consider two images I and I'. Points in image I are denoted with homogeneous coordinates $\mathtt{p} = (x, y, z)^T$, while points in image I' are denoted with homogeneous coordinates $\mathtt{p}' = (x', y', z')^T$. For the coordinates of the real world (3D) points, capital letters are used, such as $\mathtt{P} = (X, Y, Z, 1)$.

Several mathematical models describing the process of image formation exist. Here, we only give the two most commonly used.

Perspective projection model The most frequently used camera model is known as the *perspective projection model*. This corresponds to the model of an ideal pinhole camera, as illustrated in Figure 3.3. It gives a good description of the actual image formation process, as long as second order effects, such as radial distortion can be neglected. This is true for most cameras under normal acquisition circumstances, except when using small focal lengths. If the 3D coordinate axes frame is fixed to the camera, as shown in Figure 3.3 (i.e. with the origin at the projection center of the camera and the XY-plane parallel to the image plane at a distance f), perspective projection can be described as:

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z} \qquad z = 1 \tag{3.1}$$

In practice, x and y are measured in pixels instead of in the physical coordinates of the retinal plane. This is taken into account by introducing a matrix $\mathbf{K}$, usually referred to as the *calibration matrix*, which fully depends on the intrinsic camera parameters. Moreover, in general the 3D coordinate axes frame is not necessarily fixed to the camera. Hence, an additional 3D rotation and translation have to be added as well, resulting in the following formula for the perspective projection model:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \sim \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathtt{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.2}$$

with $\mathbf{R}$ a 3×3 rotation matrix, and $\mathtt{t}$ a 3×1 translation vector. $\mathbf{K}$ represents the calibration matrix and is given by:

$$\mathbf{K} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

with f_x and f_y the focal length measured in width and height of the pixels, s a factor accounting for the skew due to non-rectangular pixels and $(c_x, c_y, 1)$ the coordinates of the principal point (i.e. the projection of the camera's optical center onto the image plane).

If we make abstraction of the internal and external camera parameters ($\mathbf{K}$, respectively $\mathbf{R}$ and $\mathtt{t}$), we obtain an expression of the form

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \sim \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.4}$$

Scaled Orthographic Projection Model If the camera is sufficiently far from the scene, perspective effects are quasi nihil. In that case, the perspective projection model can be approximated by the simpler *scaled orthographic projection model*, where the object is first projected with all projection lines parallel to one another, as illustrated in Figure 3.4, followed by a scaling in the image plane. The equations for this simplified case can be derived from the perspective projection model by assuming that Z is constant, such that

$$x = mX \qquad y = mY \qquad z = 1 \qquad with \quad m = \tfrac{f}{Z} \tag{3.5}$$

For the general case, this results in a formula of the following form:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \sim \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.6}$$

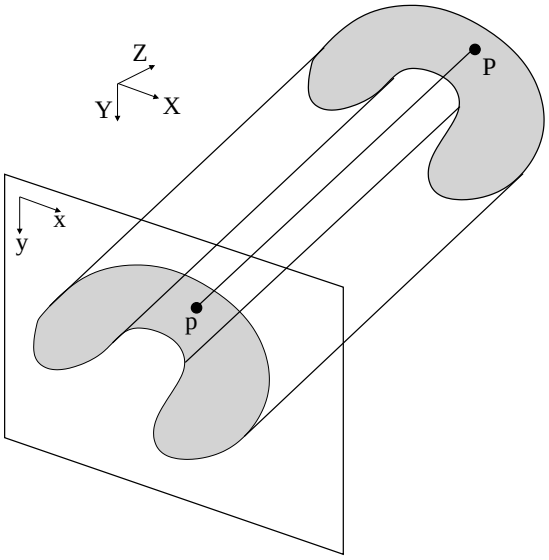


Figure 3.4: *Orthographic projection*

Planar Objects For planar objects, all 3D points lie in a plane, resulting in an extra constraint on the 3D coordinates, of the form:

$$AX + BY + CZ + D = 0 \tag{3.7}$$

This extra constraint in turn allows for simpler expressions for the transformations describing the projection of the 3D points onto the image plane. In the case of the **perspective projection model**, this results in a projective transformation or *homography*:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.8}$$

In the case of the **scaled orthographic projection model**, on the other hand, the projection is described by an *affine transformation*:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.9}$$

However, the most important thing to note is that in general this results in a *one-to-one relation* between 2D points in the image plane and 3D points in the world plane [2]. In other words, no information is lost during the image

[2]except for some degenerate cases, where the world plane projects onto a line in the image plane

projection. This greatly simplifies the mathematics, allowing easy extraction of invariants.

3.2.2 Geometric transformations between two images

Now that we have formulas at hand for describing the process of image formation, we can derive transformations describing the deformations between two images of the same rigid object or scene. Due to the easier expressions for the image formation for 2D objects, we start with this kind of objects first.

Planar objects

If the object we are looking at is known to be planar, no information is lost during projection. As a result, the transformations describing the deformations between two images are again a one-to-one relation. Different groups of transformations can be used to describe these deformations, depending on the restrictions on the motion of the camera relative to the object or vice versa. Here we start with the most general case.

- **Projective transformations**
  In the most general case, valid for any planar object and with a freely moving camera, the *perspective projection model* applies. In that case, the relation between 2D image points and 3D world coordinates can be described by a homography or projective transformation, such that the relation between the 2D image points in two different images is the combination of two such homographies, which again yields a homography.

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \sim \mathbf{H_2 H_1}^{-1} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (3.10)$$

- **Affine transformations**
  If the object is viewed from an arbitrary direction, but the distance to the object is always kept sufficiently large, the *scaled orthographic projection model* can be used instead of the more general perspective projection model. This results in an easier relation between the image coordinates as well, namely (again) an affine transformation.

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \sim \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (3.11)$$

  with $(t_x, t_y, 1)$ representing the translation parallel to the image plane, while a, b, c and d account for the rotation, skewing and scale change.

- **Similarity transformations**
  If the planar object is always viewed from a perpendicular direction, can

only rotate around an axis parallel to this direction and some basic assumptions about the internal camera parameters (such as square pixels) are fulfilled, the image deformations can be modeled by the even simpler group of *similarity transformations*, i.e. the combined effect of 2D rotation, translation and scaling.

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \sim \frac{k'}{k} \begin{pmatrix} cos(\theta) & -sin(\theta) & t_x \\ sin(\theta) & cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \tag{3.12}$$

with θ the rotation angle, $(t_x, t_y, 1)$ the translation parallel to the plane and k'/k a scale-factor to compensate for the translation orthogonal to the plane.

- **2D Rotation and Translation**
  If the distance to the object is kept fixed as well, the scale factor $\frac{k'}{k}$ in equation 3.12, drops out, leaving only 2D rotation and translation, or *Euclidean transformations.*

3D objects

For general, 3D objects, no closed formula exists linking a point in one image to the corresponding point in another image of the same object or scene. This is due to the fact that information about the depth has been lost during image projection, information which is vital for finding the corresponding point in the other image. Only in some special cases, it is possible to define a one-to-one-relation linking the two images, although the objects are not planar. This happens for instance when the camera motion consists of a pure rotation.

In general, instead of a one-to-one relation, only the *epipolar constraint* can be derived, which delineates for each point in one image a line in the other image on which the corresponding point is known to lie (and vice versa), or mathematically:

$$\mathbf{x}_2^{\mathsf{T}} \mathbf{F}\, \mathbf{x}_1 = 0 \tag{3.13}$$

with $\mathbf{F}$ a 3×3 matrix, called the *fundamental matrix* [Fau92, Har92]. If one works with metric coordinates instead of image coordinates, the *essential matrix* $\mathbf{E}$ should be used instead of the fundamental matrix $\mathbf{F}$:

$$\mathbf{x}_2'^{\mathsf{T}} \mathbf{E}\, \mathbf{x}_1' = 0 \qquad with \quad \mathbf{x}_i' = \mathbf{K_i}^{-1} \mathbf{x}_i \qquad i = 1, 2 \tag{3.14}$$

The geometrical interpretation of these equations is illustrated in Figure 3.5. Although the exact position of the 3D point `P` corresponding to a point `p` in the first image cannot be derived from the position of point `p`, `P` is known to lie on the line `L` through the optical center `C` which intersects the image plane in `p`. As a result, the exact position of the corresponding point `p'` in the second image cannot be derived either, but `p'` is known to lie on the projection of the line `L` onto the second image plane, `l'`. More generally, all the points located

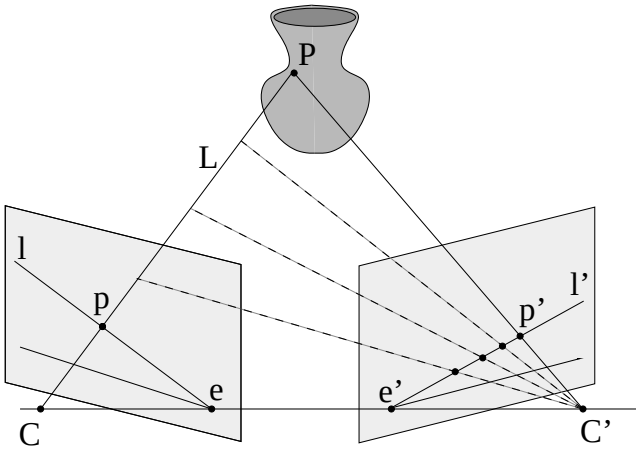


Figure 3.5: *The epipolar geometry*

on the *epipolar plane* defined by `C`, `C'` and `P` have their projections on the two *epipolar lines* `l` and `l'`. So all points on `l` have their corresponding point on `l'` and vice versa. These lines are said to be in *epipolar correspondence*.

A bundle of epipolar planes can be defined in this way, which all share the line connecting the two optical centers `C` and `C'`. Each epipolar plane yields two epipolar lines `l` and `l'`, which all intersect in `e` and `e'` respectively, the intersection points of the line `C C'` with the two image planes, coined the *epipoles*.

3.2.3 Geometric invariants

Armed with the knowledge of the transformation groups modeling the geometric deformation between two images, we can now (finally) proceed with the derivation of invariant expressions.

A transformation can be specified for different objects such as points, lines, orientations, moments, etc. A geometric configuration composed of these objects then corresponds to a set of equations. Determining the existence of invariants for this set under a given transformation is equivalent to finding a manner to eliminate the transformation parameters. When the equations are linear in the transformation parameters, the existence of invariants is guaranteed if the number of equations is greater than the number of parameters. Any polynomial transformation can be made linear by adding auxiliary dimensions or by homogenization. For more general equations, finding invariants is more complex and cannot be guaranteed.

Planar objects

Several categories of invariants exist, depending on the type of measurements used (intensities, image coordinates, derivatives, etc.). Rather than deriving one particular set of invariants, we give an overview of these different classes, more or less in chronological order, together with some examples for each type summarized in Table 3.1.

- **Moment invariants**
  Invariants were first used in the context of computer vision by [Hu62], in the form of *moment invariants*. The basic measurement consists of *moments*, which integrate powers of the image coordinates and possibly intensity information over a region:

$$MS_{pq} = \iint_{\Omega} x^p y^q \, w(x,y) \, dxdy \tag{3.15}$$

$$MI_{pq} = \iint_{\Omega} x^p y^q \, I(x,y) \, w(x,y) \, dxdy \tag{3.16}$$

  with Ω the region over which the intensity and coordinates have been integrated and $w(x,y)$ a weighting function.

  Different kinds of moments have been defined (Legendre, Zernike, rotational, etc.). However, in this work we will use the most traditional moments, sometimes referred to as *geometric* moments, with $w(x,y)$ the identity function, as these are the least sensitive to noise [TC88].

  More precisely, we use a variation on these geometric moment invariants, introduced recently by Mindru *et al.* [MMVG98]. These are so-called *Generalized Color Moment Invariants*, based on generalized color moments:

$$M_{pq}^{abc} = \iint_{\Omega} x^p y^q [R(x,y)]^a [G(x,y)]^b [B(x,y)]^c \, dxdy \tag{3.17}$$

  As they combine information from several spectral bands and also use powers of intensities, the same degree of invariance can often be obtained with lower order moments.

- **Algebraic invariants**
  Algebraic invariants start from a combination of simple geometric entities such as points, lines or conics [SRP95, MZ92]. The examples given in table 3.1 represent respectively the area spanned by three points (invariant under 2D rotation and translation), the ratio of two such areas (affinely invariant) and the cross-ratio of four collinear points (projectively invariant).

However, the difficulty of characterizing an arbitrary object with such simple geometric structures has constrained the use of algebraic invariants to specific (mostly industrial) classes of objects. Also, the small number of these invariants often fails to provide sufficient discriminative capability when the amount of objects increases.

- **Differential invariants**
  Differential invariants, as introduced by [Wei88, CygAl87], have been designed to generalize previous approaches to a larger class of objects. Differential geometry looks at a curve as the graph of a function, which is completely characterized by its local behavior (bending, twisting, etc.) at each of its points. This local behavior is captured in terms of derivatives of the curve with respect to some parameter. Advantages of differential invariants include locality and the coverage of a relatively large class of shapes, at the cost of a higher sensitivity to noise through the use of (higher order) derivatives, especially in case of affine or projective invariance.

- **Semi-differential invariants**
  Semi-differential invariants have then been introduced as a combination of algebraic and differential invariants, trying to combine their advantages while avoiding their drawbacks [VGAl90, BriAl92, MooAl95]. In particular, such invariants are based on a combination of reference points and coordinate derivatives. The important thing is that less reference points than needed for the calculation of algebraic invariants are combined with lower orders of differentiation than needed for the calculation of differential invariants.

The above list is by far not exhaustive. Many other invariant constructions exist. Invariants have been derived based on Gaussian derivatives, reflectance ratios, and so on. For an overview on geometric invariance, we refer to [MZ92, MZ94, Woo96].

There is no guarantee however that invariants exist for all cases. Projectively invariant moments for instance cannot be found.

3D objects

No general invariants exist when one considers 2D image points produced by an arbitrary set of 3D scene points [BWR90, CJ91]. The only way out is to increase the number of measurements by using several images simultaneously (with known correspondences !), or to decrease the dimensionality of the transformations by making some simplifying assumptions about the scene (e.g. planar objects or surfaces of revolution) or about the camera motion (e.g. only rotations).

In this work, we use the fact that most 3D objects or scenes can *locally* be approximated by a planar surface, so we can use the 2D-invariants derived

Table 3.1: *Examples of different types of invariants.*

2D Rotation and Translation

Moment inv.	$(\frac{M^1_{10}M^0_{00} - M^0_{10}M^1_{00}}{M^1_{00}M^0_{00}})^2 + (\frac{M^1_{01}M^0_{00} - M^0_{01}M^1_{00}}{M^1_{00}M^0_{00}})^2$
Algebraic inv.	$\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_3 \rvert$
Differential inv.	$\frac{\lvert \mathbf{x}_1^{(1)} \quad \mathbf{x}_1^{(2)} \rvert}{\lVert \mathbf{x}_1^{(1)} \rVert^3}$
Semidifferential inv.	$\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1^{(1)} \rvert$

Affine Transformations

Moment inv.	$\frac{\begin{pmatrix} M^1_{10}M^2_{01}M^0_{00}M^0_{00} - M^0_{10}M^2_{01}M^1_{00}M^0_{00} \\ -M^1_{10}M^0_{01}M^2_{00}M^0_{00} + M^0_{10}M^0_{01}M^1_{00}M^2_{00} \\ -M^1_{01}M^2_{10}M^0_{00}M^0_{00} - M^0_{01}M^2_{10}M^1_{00}M^0_{00} \\ -M^1_{01}M^0_{10}M^2_{00}M^0_{00} + M^0_{01}M^0_{10}M^1_{00}M^2_{00} \end{pmatrix}}{M^1_{00}M^2_{00}M^0_{00}M^0_{00}}$
Algebraic inv.	$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_3 \rvert}{\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_4 \rvert}$
Semidifferential inv.	$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1^{(1)} \rvert}{\lvert \mathbf{x}_1 - \mathbf{x}_3 \quad \mathbf{x}_1^{(1)} \rvert}$

Projective Transformations

Moment inv.	–
Algebraic inv.	$\frac{(x_1 - x_2)(x_3 - x_4)}{(x_1 - x_4)(x_2 - x_3)}$
Semidifferential inv.	$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_2^{(1)} \rvert^3 . \lvert \mathbf{x}_1^{(1)} \quad \mathbf{x}_1^{(2)} \rvert}{\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1^{(1)} \rvert^3 . \lvert \mathbf{x}_2^{(1)} \quad \mathbf{x}_2^{(2)} \rvert}$

in the previous paragraph on a local scale. Most three-dimensional objects or scenes will indeed have surfaces that smoothly vary in a continuous way, such that the deviations from a local, first-order (i.e. planar) approximation will be limited as long as we only consider a small patch of the surface. Near the borders of an object, or near a (self)occlusion, the planarity assumption will be violated, but otherwise this seems a reasonable assumption. In fact, the 2D invariants can be considered as *quasi-invariants* in case of 3D objects [BL93].

Jacobs [Jac97] has proposed another solution, in an attempt to generalize the idea of invariance for 3D to 2D matching. He has shown that, under scaled orthographic projection, a measure can be derived based on a set of 2D points, that is not invariant, but that is known to lie on two lines in two high-dimensional spaces, and that these lines are the lowest dimensional representation possible. The parameters of these lines have to be determined based on the 3D model. Then, matching can be accomplished by checking whether the image point set yields a measure close to the lines or not.

3.3 Photometric invariance

Apart from the geometric deformations in the image, also photometric changes have to be taken into account. These are the result of changes in the illumination conditions or changes in the relative orientation of the local surface with respect to the light source and camera. While the geometric deformations can be modeled quite accurately (as long as the surface can be assumed to be planar), photometric deformations largely depend on the material of the object and are hard to capture in a mathematical model.

3.3.1 Photometric transformations involved in image formation

To understand the changes in intensity or color of a point in an image due to different illumination conditions or changes in the relative orientation of the surface with respect to the camera or light source, we first have to gain some insight in the physics of surface reflection. Two mathematical models exist to describe how a surface reflects the incident light: the model of specular reflection and the model of a diffuse reflection (see Figure 3.6). *Specular reflection* works like the reflection in a mirror. The light is reflected at an angle to the surface normal equal to the angle of the incident light. In the case of *diffuse reflection*, on the other hand, the light is reflected equally in all directions. A surface with only diffuse reflection is usually referred to as a *lambertian surface*. In practice, the reflection of a surface is usually a combination of these two models, depending on the surface material. Metal for instance has a large portion of specular reflection while paper typically is more or less lambertian.

The lambertian portion of the reflection does not depend on the relative orientation of the surface with respect to the camera, and hence is easier to model

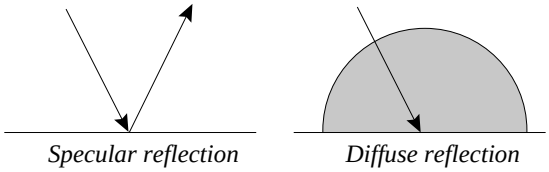


Figure 3.6: *Specular and diffuse reflection models*

and less dependent on changes in viewpoint. Specular reflection, on the other hand, only occurs at a specific angle, where it changes the surface appearance drastically. As the geometry is in general not known or controlled, specular reflection will not be taken into account in our reflection model. Instead, we assume that it happens only occasionally and on a local scale, such that it can be considered a local disturbance.

In case of a lambertian surface, the intensity of a specific pixel in the image depends on the following components: the spectral properties of the object's surface (i.e. the surface material and color), the spectrum or color of the incident light, the camera sensitivity and the relative orientation of the surface with respect to the incident light. These are combined in the following way:

$$I = \mathtt{e.n} \int_{\omega} S(\lambda)E(\lambda)F_i(\lambda)d\lambda \tag{3.18}$$

with I the intensity (i.e. R, G or B), λ the wavelength, $\mathtt{e}$ the orientation of the incident light, $\mathtt{n}$ the surface normal, $S(\lambda)$ the surface reflectance, $E(\lambda)$ the spectrum of the light source and $F_i(\lambda)$ the response function of the camera for the i-th color-band.

The above expression can further be simplified if we can assume that the response function $F_i(\lambda)$ corresponds to a narrow bandwidth filter. This is more or less true for most commercially available cameras. If not, extra filters can be added to fulfill this constraint. In that case, equation 3.18 becomes:

$$I = \mathtt{e.n}\, S(\lambda_i)E(\lambda_i)F_i(\lambda_i) \tag{3.19}$$

3.3.2 Photometric transformations between two images

The assumptions that the camera is relatively far from the object and that the object is locally planar greatly simplify the analysis of the photometric changes. Typically light sources are far from the objects as well. The geometry of light reflection is the same for all points on a plane in that case, i.e. they share the same angles of light incidence and camera viewing direction.

Using Equations 3.18 and 3.19, it can be shown that for a lambertian surface a change in position of the light source results in an overall scaling of the intensities [ON94] (same scale-factor for R, G and B). A change in the illumination color (spectrum of the light source) on the other hand corresponds

to a different scalefactor for each of the three color-bands (if one can assume that the response functions of the camera can be modeled as narrow bandwidth filters). So in case of lambertian reflection, a different scaling factor for each spectral band suffices to model the effects of changing illumination.

To this diagonal model, we add an additional offset for each spectral band. This has been shown to better model the combined effect of diffuse and specular reflection [Wol94] and to give better performance [Rei93]. The three offsets are taken different to account for the property that specularly reflected color depends more on the color of the light source than on the color of the surface [KSK88]. This results in the following model for the changes in image intensities between two images of the same object or scene:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} s_r & 0 & 0 \\ 0 & s_g & 0 \\ 0 & 0 & s_b \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_r \\ o_g \\ o_b \end{pmatrix} \tag{3.20}$$

3.3.3 Photometric invariants

As a first example of photometric invariance, we refer to *reflectance ratios*, as used in the recognition work of [NB96, NG98].

$$\rho = \frac{R_1 G_2}{R_2 G_1} \tag{3.21}$$

However, these only compensate for the scale-factors in Equation 3.20, not for the offsets. In this work, we use photometric invariants based on generalized color moments [MMVG98]. An example of such an invariant that can cope with both the scalefactors and offsets is:

$$\frac{\left[M_{00}^{110} M_{00}^{000} - M_{00}^{100} M_{00}^{010}\right]^2}{\left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2\right] \left[M_{00}^{020} M_{00}^{000} - (M_{00}^{010})^2\right]} \tag{3.22}$$

As fully invariant expressions can become rather complex, the offsets are sometimes compensated for in a normalization step prior to the actual invariant computation, e.g. by using the intensity minus average intensity or the absolute value of this difference.

3.4 Local, affinely invariant features

One of the goals of this work consists of reducing the number of reference images needed for object recognition, by increasing the level of invariance from 2D rotation and translation (as used by many state-of-the-art recognition systems) to invariance under plane affine transformations.

Apart from being invariant, we want our features to be *local* as well. As already mentioned in chapter 2, local features make a system robust to occlusions, image clutter and changing backgrounds and avoid the need for segmentation.

Also, staying local increases the probability of planarity, which we need in order to use the 2D invariants as quasi-invariants for the 3D case.

However, locality and increased levels of invariance are hard to combine. Every measurement – be it intensities, derivatives, moments or whatever – is inherently computed based on a specific *support* (implicitly or explicitly) [3]. In order to keep the invariance intact, one must make sure that the same support is used for the invariant computations before and after the transformation.

As long as the geometric deformations are limited to 2D rotation and translation, this is not a problem. By choosing a circular neighborhood for the support (e.g. using isotropic Gaussian smoothing kernels), one can make sure that the features in different images are computed based on the same support. However, there is no straightforward extension of this approach to changes in scale or affine transformations. The problem is that there are no regions with fixed shape and size under this wider group of transformations. Circular regions for instance will be deformed to ellipses of different shapes and sizes.

In the case of scale changes, methods to automatically select an appropriate scale, in an invariant way (i.e. such that corresponding scales are found in two different images) have already been proposed [Lin93, Lin98, Low99]. In this work, we develop something similar for the more general case of affine transformations. The idea is to come up with region extraction methods that allow to automatically select local neighborhoods in such a way that similar regions are found in different images, (i.e. regions that cover the same physical part of the object irrespective of the viewpoint). We have coined these special neighborhoods that survive affine transformations *affinely invariant regions*. The crux of the matter is that these shapes are determined within a single image, without any information about the other views that might have been taken. This is essential, as no correspondences are available yet, and we want to avoid the need for time-consuming combinatorics.

An example of such affinely invariant regions, as extracted by one of the methods proposed in this work, is given in Figure 3.7. Although the two views differ substantially in their respective viewpoints (a), the example regions shown in (b) cover the same part of the scene. Their shapes have been extracted automatically and independently from each other.

3.5 Conclusion

For mathematical reasons, invariants are obtained more easily for *planar* geometric structures. This has initially limited their use to 2D objects, such as object facets and trademark recognition. While two-dimensional invariant feature extraction is widely applied nowadays, invariants for three-dimensional objects are still an open research domain. Indeed, as the geometric deformations cannot be modeled by a group of transformations in this case, a straightforward

[3] With *support* we refer to the physical part of the (3D) object or scene whose projection is used in the computation of the measurement.

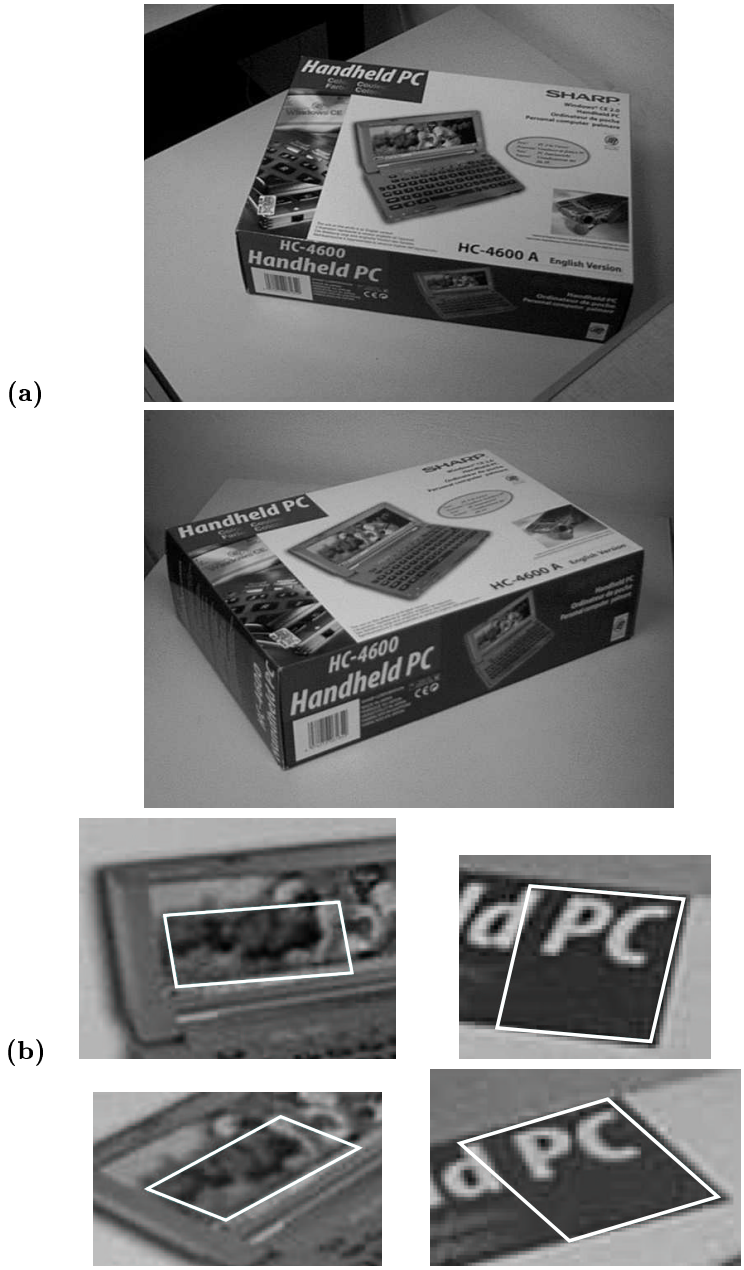


Figure 3.7: **(a)** *Two images of the same object.* **(b)** *Two parallelogram-shaped patches as they are generated by the system: when the viewpoint changes the shapes of the patches are transformed automatically so that they always cover the same physical part of the scene.*

extension from 2D to 3D seems impossible. Instead, quasi-invariants are often used. These are features that are only slightly influenced by the considered transformation instead of staying completely unaltered.

In this work, we assume that a 3D object surface can *locally* be approximated by a plane, such that on a local scale, the 2D transformation groups and 2D invariants that come with it can be used. Also, it is assumed that the distance between camera and object is sufficiently large (compared to the local, relative distances), such that perspective deformations can be neglected. Under these conditions, an arbitrary change in viewpoint can be modeled by a plane affine transformation.

The effect of changes in the lighting conditions on the other hand is described using a linear model, with different offsets and different scale factors for each of the three color bands.

Achieving this high level of invariance on a local scale in a robust way is not straightforward though. Algebraic invariants (based on combinations of points) are no longer local. Differential invariants need higher order derivatives and as a result are not stable enough as they depend on the image support over which they have been computed and are sensitive to noise, while semi-differential invariants based on edge segments tend to be stretched out over too large a distance before gaining sufficient discriminative power, again jeopardizing the locality-assumption. It is our belief that the best way to stay local *and* gain sufficient discriminative power is to directly exploit the intensity profile in a small neighborhood around a point, through the use of moment invariants.

However, the largest problem is that no neighborhood with fixed shape exists that remains unaltered under an affine transformation. A circular region, for instance, is transformed into an elliptic one. What we need is some kind of self-adaptive regions, that deform their shape based on the underlying image intensities as to keep on representing the same physical part of an object under changing viewpoint, similar to the automatic scale detection developed by [Lin98]. We will refer to this special kind of neighborhoods as *affinely invariant regions*. How one can derive such affinely invariant regions will be explained in the next chapter.

Chapter 4

Extracting local, affinely invariant regions

This chapter is organized as follows. First, after a short introduction (section 4.1), the selection of good anchor points is discussed in section 4.2. These are points where the construction of the invariant regions stands a good chance of being successful. When looking for affinely invariant regions, the search is focused on the neighborhoods around such anchor points. Next, two different methods for extracting local, affinely invariant regions are proposed in sections 4.3 and 4.4. The first method makes use of geometric entities such as corners and edges to guide the region extraction process, while the second method is directly based on the intensity profile. The final goal is to obtain an opportunistic system that exploits several types of invariant regions simultaneously. In section 4.5, the two different methods are evaluated and compared. Section 4.6 concludes the chapter.

4.1 Introduction

This chapter describes how we construct the affinely invariant regions. We are after affine invariance as we assume that the regions cover surface patches that are more or less planar. By keeping the regions reasonably small, relative deformations between views for such patches can be modeled well as affine transformations, since perspective deformations can be neglected on a small scale. Apart from the geometric invariance, photometric invariance should be included as well. In this context, we allow changes in the colors in as far as they can be modeled as independent scalings and offsets for each of the three color bands. As explained in the previous chapter, *affinely invariant regions* are a necessary tool to arrive at this high degree of invariance on a local scale. These are self-adapting regions, that change their shape depending on the underlying intensity profile as to keep on representing the same physical part of an object.

As such, they allow to compute invariants in different images based on the same support. The crux of the matter is that these regions should be extracted for each image separately, i.e. without using any a priori knowledge about the corresponding region in another image. This is important, as no correspondences are available yet and we want to avoid time consuming combinatorics.

At first sight, extracting such affinely invariant regions seems a rather straightforward task. In theory, one can for instance use the following equation for an affinely invariant elliptic region around an anchor point $\mathtt{x}_0$ based on the second order derivatives [1]:

$$(\mathtt{x}-\mathtt{x}_0)^T \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} (\mathtt{x}-\mathtt{x}_0) = 1 \qquad (\mathtt{x}, \mathtt{x}_0 \in I\!R^2) \qquad (4.1)$$

In practice however, the second order spatial derivatives are not stable. They too are sensitive to the affine deformations, as $I(x, y)$ is not a continuous function but rather discretized into pixels, such that the derivatives are computed using discrete approximations (of fixed size). As a result, their *support* changes when changing the viewpoint or scale and so do their values. In fact, this is a chicken-and-egg problem: only after having found a local, affinely invariant region, the derivatives can be computed in a stable way by adapting the size of the discrete approximation mask accordingly (e.g. using the projection onto non-uniform Gaussian derivatives).

Nevertheless, some attempts to extract elliptic regions based on Equation 4.1 or variations thereof can be found in the literature (see chapter 2), mainly in the context of texture analysis. Typically, an iterative scheme is applied, in combination with some serious restrictions on the intensity profile to ensure convergence. Ballester and Gonzales [BG98] work with regular texture images, while the method of Gårding and Lindeberg [GL94] assumes nice, well-behaving blobs in the image.

The methods we propose here mostly differ from these, in that we do not apply an iterative procedure, and hence we do not have to limit ourselves to a specific kind of images or regions to guarantee convergence (although we cannot guarantee to find a region for every anchor point either).

In fact, we do not believe that it is possible to find a single, universal type of regions (or any other feature for that matter) that is applicable to the wide diversity of images encountered in every day life. Instead of trying to find the 'optimal' region extraction method that should work in all cases, we follow a more heuristic approach. Our goal is to develop an *opportunistic system* that exploits a wide diversity of regions simultaneously, depending on what is on

[1] Note that this method is only geometrically invariant. Under photometric changes (three different scalings and offsets), the size of the ellipse can still vary, although its shape (orientation and eccentricity) remains unaltered. This can be solved e.g. by adding an extra automatic scale detection step [Lin98].

offer in a specific image. This should increase the robustness, both with respect to the applicability of our method and with respect to the number of correspondences found. Other kind of regions than those proposed in this work can be added as well, both general ones as more specific ones tuned towards specific features or applications. Some ideas for alternative construction methods are given in section 4.5. The more region extraction methods available to the system, the higher the number of corresponding regions found and the more accurate and robust the resulting object recognition (or any other application).

4.2 Selection of anchor points

The first step in the extraction of affinely invariant regions consists of selecting *anchor points*, around which the actual regions will be constructed. This allows to reduce the needed computation time, since the attention can be focused on these points instead of examining every single pixel in the image. Moreover, the complexity of the problem can be reduced as well, since extra assumptions can often be made concerning the local neighborhood based on the type of anchor point used.

Good anchor points are points that result in stable invariant regions, are repeatable and easy to detect. With repeatability, we mean that there is a high probability that the same point will be found in another view as well (or at least, a point that results in the same region). In this work, we use two different types of anchor points: Harris corner points and local intensity extrema.

4.2.1 Harris corner points

The first type of anchor points we use are corner points, as detected by the Harris corner detector [Har88]. This method selects points with an intensity profile that shows quite some changes and a substantial bending orthogonal to the gradient direction. As such, not only real corners, such as the corner of a room or a table top, are selected, but also T-junctions, endpoints of a line, points on an edge with high curvature and so on.

Their detection is based on the evaluation of a matrix C which represents the local statistics of the first order derivatives of image brightness $I(x, y)$ with respect to both spatial image coordinates x and y.

$$\mathbf{C} = \begin{pmatrix} \widehat{(\frac{\partial I}{\partial x})^2} & \widehat{\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}} \\ \widehat{\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}} & \widehat{(\frac{\partial I}{\partial y})^2} \end{pmatrix} \tag{4.2}$$

where $\widehat{x}$ signifies averages of the value x over a small neighborhood around the point. The average typically is not a simple mean value, but is based on weighting the values at neighboring positions with a Gaussian fall-off function. Points are selected for which

$$|\mathbf{C}| - k\,(trace(\mathbf{C}))^2 \geq \delta_H \tag{4.3}$$

with k typically set to 0.04, $|\ldots|$ representing the determinant of a matrix and δ_H a threshold. As the determinant of a matrix is equal to the product of the eigenvalues and the trace of a matrix is equal to the sum of the eigenvalues, the above expression will typically give high values if and only if both eigenvalues are large. This can be interpreted as large variations in $I(x, y)$ in two dimensions.

Harris corner points are good candidates for our anchor points. They are not really affinely invariant, as the support over which the averages in $\mathbf{C}$ are computed is not adapted to the affine deformations. Nevertheless, the resulting points mostly survive the affine transformations. This can be explained by the fact that no quantitative information (e.g. the exact value of the left hand side expression of equation 4.3) is used in their extraction, only qualitative information. In a recent comparison of several interest point detectors [SM00], they obtained the highest score with respect to repeatability (i.e. robustness to viewpoint and illumination changes). Another advantage, as shown in the same study, is that Harris corner points typically contain a large amount of information, resulting in a high discriminative power, which facilitates the matching of regions between images.

It is often argued that Harris corner points are also very precisely localized, even up to sub-pixel accuracy. However, this sub-pixel accuracy quickly vanishes under larger changes in viewpoint or scale. Still, for our purpose this sub-pixel accuracy is not vital.

The main drawback of using Harris corner points as anchor points for our affinely invariant region extraction is of a more practical nature. Being corners, they often tend to lie near the border of an object, close to a depth discontinuity. Hence, the planarity assumption is often violated in the immediate neighborhood of such a corner point. We will come back to this point later.

Parameters

In this work, we used the implementation of the Harris corner detector supplied by the TargetJr software distribution [TJr]. The main parameter that has to be specified is an approximation of the number of corners to be found in the image, which is 300 by default, but may have to be adapted depending on the size of the image or the image content.

For the default values for the other parameters, we refer to appendix C.

Example

Figure 4.1 shows the Harris corner points detected in two sample images of the same object, taken from different viewpoints. Clearly, many of the corners found in the left image have been found in the right image as well.

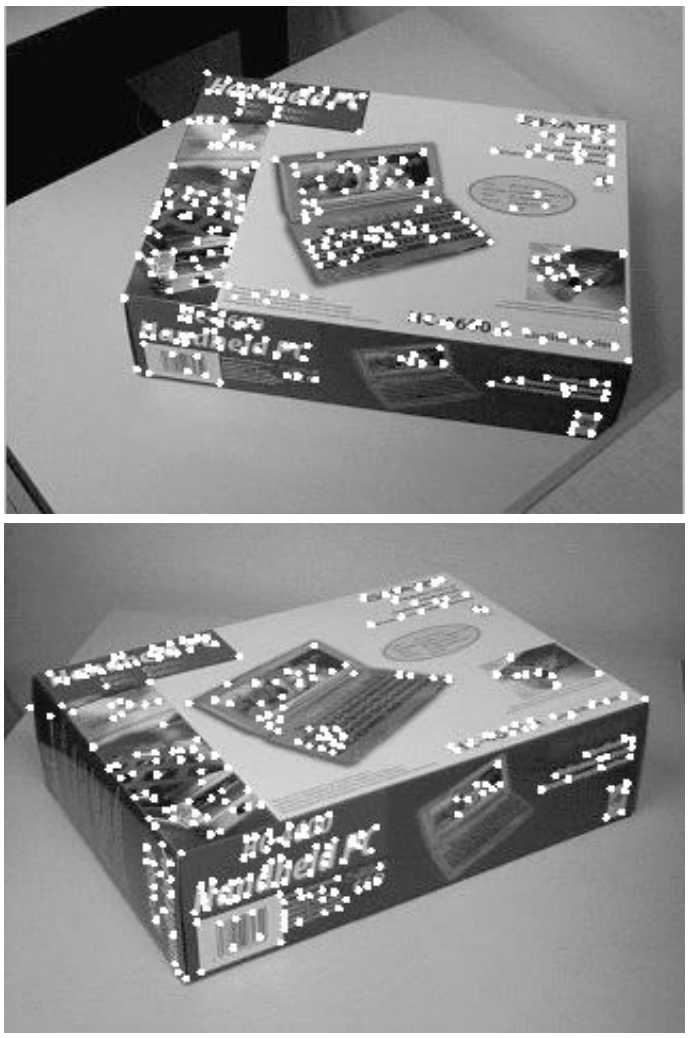


Figure 4.1: *Harris corner points extracted in two sample images of the same object taken from two different viewpoints.*

4.2.2 Local intensity extrema

Instead of using corners as anchor points, one can also start from local extrema in the image brightness $I(x, y)$. To this end, we first apply some Gaussian smoothing to the image to reduce the effect of noise, which otherwise causes too many unstable local extrema. Then, the local extrema are extracted with a non-maximum suppression algorithm. An anchor point is created at pixel (x_a, y_a) if there is no other pixel (x, y) with $|x - x_a| \leq \delta$ and $|y - y_a| \leq \delta$ for which $I(x, y) \leq I(x_a, y_a)$ (in case of a local minimum) or $I(x, y) \geq I(x_a, y_a)$ (in case of a local maximum).

These points cannot be localized as accurately as Harris corner points, since the local extrema in intensity are often rather smooth. However, they can withstand any continuous geometric deformation and any monotonic transformation of the intensity. Also, they are less likely to lie near the border of an object resulting in a non-planar neighborhood, a major drawback when working with Harris corner points. In practice, local intensity extrema often correspond to the centers of extra light or extra dark 'blobs' in the image.

Parameters

The parameters to be specified for the extraction of the local intensity extrema are the sigma of the Gaussian smoothing kernel (default value 3) and the window size δ used for the non-maximum suppression (default value 10).

Example

An example of the local intensity extrema found in two sample images of the same object taken from different viewpoints can be found in Figure 4.2. The locations in the two images are not always exactly the same, but in many cases, intensity extrema can be found at more or less the same positions.

4.2.3 Comparison

Which type of anchor points performs best also depends on the method used for the invariant region extraction, how well this method knows to exploit the particular characteristics of the anchor points and how well it can deal with the shortcomings.

For instance, in the neighborhood of a Harris corner point, one can often detect one or more edges, as there are large changes in intensities. These edges can be exploited in the region extraction process. At the same time, the high risk for non-planarity due to the high probability of a corner point lying near the border of an object can be alleviated by constructing a region that is not centered around the corner point, but deligned by the edges (representing the border of the object).

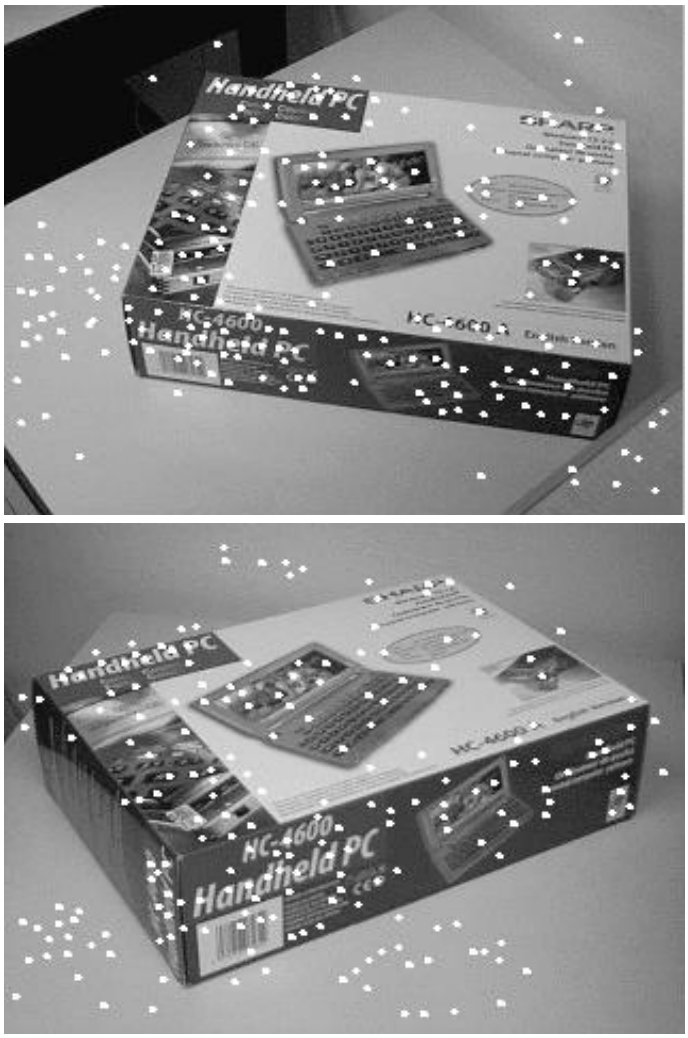


Figure 4.2: *Local intensity extrema extracted in two sample images of the same object taken from different viewpoints.*

Similarly, regions starting from local intensity extrema should not depend too much on the exact position of the extrema, to overcome the inaccurate localization of these points.

Other types of anchor points are possible as well. For instance, Lowe [Low99] uses extrema over scale space of a difference of Gaussians (DOG) filter.

4.3 Geometry-based region extraction

The first method for affinely invariant region extraction we develop starts from Harris corner points (see section 4.2.1). Apart from the corners, it also exploits the edges that can usually be found in the neighborhood of such a point. As it relies to such a high degree on the geometric entities present in the image (corners and edges), we coined this method *geometry-based* region extraction.

The edges are detected using an implementation of the Canny edge detector [Can86], which, roughly speaking, detects edges at points that are local maxima of the first derivative of the smoothed image in the direction of the gradient (with non-maximum suppression) and progressively promotes weak edges to strong edges if they are contiguous with a strong edge.

Depending on the nature of the edges in the neighborhood of the corner point, two different cases are distinguished: one method is developed for curved edges while a slightly different method is applied in case of straight edges. The distinction between both is made on the basis of the result of a simple line fitting algorithm.

4.3.1 Case 1: Curved edges

Let $\mathtt{p} = (x_p, y_p)$ be a Harris corner point, and e_1 and e_2 two edges in its neighborhood. Then two relative affinely invariant parameters l_1 and l_2 can be defined for the two edges e_1 and e_2, in the following way (see also Figure 4.3)

$$l_i = \int abs(\ |\mathtt{p_i}^{(1)}(s_i)\ \ \mathtt{p} - \mathtt{p_i}(s_i)|\ )\, ds_i \qquad i = 1, 2 \tag{4.4}$$

with s_i an arbitrary curve parameter, $\mathtt{p_i}^{(1)}(s_i)$ the first derivative of $\mathtt{p_i}(s_i)$ with respect to s_i, $abs()$ the absolute value and $|..|$ the determinant. Then, each point $\mathtt{p_1}(l_1)$ on one edge can be associated with a point $\mathtt{p_2}(l_2)$ on the other edge, such that $l_1 = l_2$. Both l_1 and l_2 are relative affine invariants, such that their ratio l_1/l_2 is an absolute affine invariant and the association of a point on one edge with a point on the other edge is also affinely invariant. From now on, we will simply use l when referring to $l_1 = l_2$.

Together, the two points $\mathtt{p_1}(l)$ and $\mathtt{p_2}(l)$ define a region $\Omega(l)$ for the point $\mathtt{p}$ as a function of l: the parallelogram spanned by the vectors $\mathtt{p_1}(l) - \mathtt{p}$ and $\mathtt{p_2}(l) - \mathtt{p}$ (see Figure 4.3). In this way, the problem of finding an affinely invariant region has been reduced to selecting an appropriate value for l in an affinely invariant way.

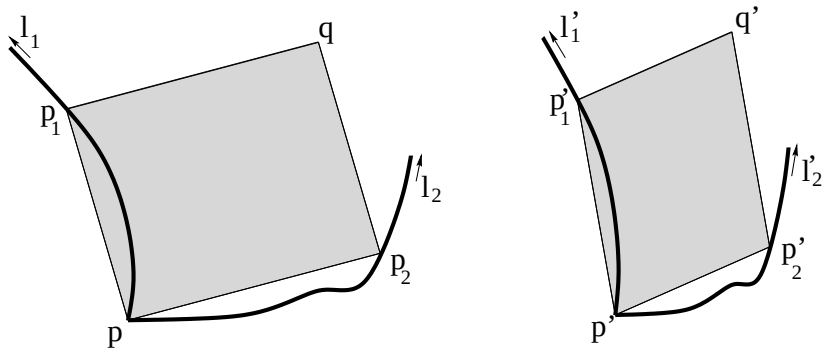


Figure 4.3: *Based on the edges in the neighborhood of a corner point* p*, an affinely invariant region can be defined as a function of the relative affinely invariant parameter* $l_1 = l_2$.

Next, we evaluate a function over the region $\Omega(l)$ that reaches its extrema in an invariant way for both the geometric and photometric changes. We then select the region(s) $\Omega(l)$ for which the function reaches such a local extremum. We typically generate regions for a few extrema, which introduces a kind of scale concept as now regions of different sizes coexist for a single corner. Since it is not guaranteed that a single function will reach an extremum over the limited l-interval we are looking at, a set of functions is tested. Taking extrema of several functions into account, we get a better guarantee that a high number of corners will indeed generate some regions. Three examples of such functions are:

$$f_1(\Omega) = \frac{M^1_{00}}{M^0_{00}} \tag{4.5}$$

$$f_2(\Omega) = abs\left(\frac{\begin{vmatrix} \mathtt{p}_1 - \mathtt{p}_g & \mathtt{p}_2 - \mathtt{p}_g \end{vmatrix}}{\begin{vmatrix} \mathtt{p} - \mathtt{p}_1 & \mathtt{p} - \mathtt{p}_2 \end{vmatrix}}\right) \frac{M^1_{00}}{\sqrt{M^2_{00} M^0_{00} - (M^1_{00})^2}} \tag{4.6}$$

$$f_3(\Omega) = abs\left(\frac{\begin{vmatrix} \mathtt{p} - \mathtt{p}_g & \mathtt{q} - \mathtt{p}_g \end{vmatrix}}{\begin{vmatrix} \mathtt{p} - \mathtt{p}_1 & \mathtt{p} - \mathtt{p}_2 \end{vmatrix}}\right) \frac{M^1_{00}}{\sqrt{M^2_{00} M^0_{00} - (M^1_{00})^2}} \tag{4.7}$$

$$with \quad M^n_{pq} = \int_\Omega I^n(x,y) x^p y^q \; dxdy$$

$$\mathtt{p}_g = \left(\frac{M^1_{10}}{M^1_{00}}, \frac{M^1_{01}}{M^1_{00}}\right)$$

with M^n_{pq} the nth order, $(p+q)$th degree moment computed over the region $\Omega(l)$, $\mathtt{p}_g$ the center of gravity of the region, weighted with $I(x,y)$ (either R, G or B), and $\mathtt{q}$ the corner of the parallelogram opposite to the corner point $\mathtt{p}$ (see Figure 4.3).

The first function represents the average intensity over the region $\Omega(l)$. It is just given because of its simplicity, and as an example of a function that

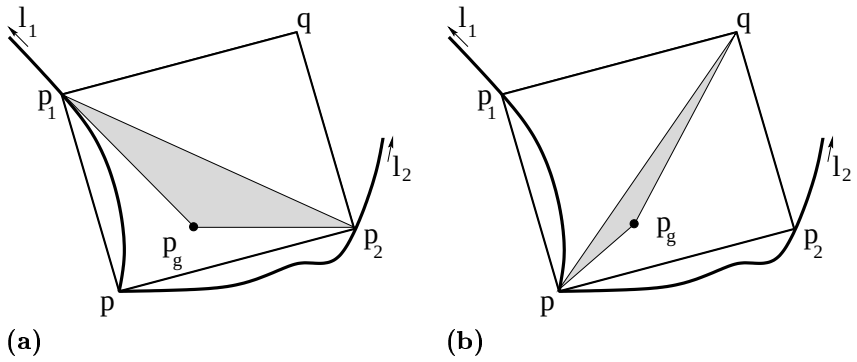


Figure 4.4: *Geometrical interpretation of the functions* $f_2(\Omega)$ *(***a***) and* $f_3(\Omega)$ *(***b***).*

is not in itself invariant but reaches its extrema in an invariant way (i.e. the region Ω^* for which $f_1(\Omega)$ reaches an extremum is invariant, but the value of $f_1(\Omega^*)$ is not). In contrast to $f_1(\Omega)$, $f_2(\Omega)$ and $f_3(\Omega)$ are in themselves already invariant. Nevertheless, also for $f_2(\Omega)$ and $f_3(\Omega)$, we still select the points where the functions reach an extremum, as this is probably more robust then simply selecting points where the function reaches a specific value.

In practice, $f_2(\Omega)$ and $f_3(\Omega)$ yield much better, more outspoken extrema, and hence much more stable regions than $f_1(\Omega)$. They consist of two components: first, the ratio of two areas, one of which depends on the center of gravity weighted with intensity and hence on the region pattern, and second, a factor which compensates for the dependence of the first component to offsets in the image intensity. Alternatively, one could leave out this second factor, and compensate for the photometric offsets by an appropriate normalization of the intensities before computing the moments [2].

Figure 4.4 shows the geometrical interpretation of the first component of $f_2(\Omega)$ (left) and $f_3(\Omega)$ (right). It's twice the ratio of the marked area, divided by the total area of the region. By looking for local minima of these functions we favor more *balanced* regions, i.e. regions for which the center of gravity lies on or close to one of the diagonals of the area.

For a proof of the geometric and photometric invariance of the local extrema of $f_1(\Omega)$, $f_2(\Omega)$ and $f_3(\Omega)$, we refer to appendix A.

[2] Note that a normalization of the form $I'(x,y) = I(x,y) - I(\widehat{x,y})$ cannot be used in this case, as this would result in $M^1_{00} = 0$ and hence a division by zero in the computation of the center of gravity. A normalization of the form $I'(x,y) = abs(I(x,y) - I(\widehat{x,y}))$ will do the job, but at the cost of reducing the range of the intensity values by approximately half.

Implementation details

The implementation of this first region extraction method is rather straightforward, except maybe for the evaluation of the relative affinely invariant parameters l_1 and l_2 (see Equation 4.4), as these use first order derivatives along the edges. To this end, a spline is fitted to the edge, allowing easy and accurate computation of derivatives.

In practice, only local *minima* of $f_2(\Omega)$ and $f_3(\Omega)$ are used, as in practice they turn out to be localized more accurately than the local maxima.

Internally, an affinely invariant region is represented as a combination of three points: $\mathtt{p}$, $\mathtt{p}_1$ and $\mathtt{p}_2$ (see Figure 4.3). These three points can be used as an affine basis for describing the intensity profile within the region. $\mathtt{p}_1$ and $\mathtt{p}_2$ are chosen in such a way that the angle from $\overline{\mathtt{p}\,\mathtt{p}_1}$ to $\overline{\mathtt{p}\,\mathtt{p}_2}$ in clockwise direction is smaller than 180 degrees.

Parameters

The most important parameters that have to be specified in order to extract geometry-based regions in case of curved edges are summarized here.

- **Canny edge parameters**
  In this work, we used the implementation of the Canny edge detector supplied by the TargetJr software distribution [TJr]. The parameters for the Canny edge detection stay unaltered at their default values most of the time. For images of very poor quality, it could be advisable to increase the sigma of the Gaussian used for the detection, or to lower the minimum intensity difference over an edge.

- **Region size**
  Two other important parameters specify the minimum and maximum region size. These values delimit the search area over which local extrema of the functions are retained. Too small regions are not determined accurately as they depend too much on the discretization. Moreover, the intensity profile within a region of a few pixels will not be sufficient to discriminate it. Increasing the maximum region size increases the number of regions found and the robustness of the system to large scale changes. On the other hand, it also increases the computation time, and the larger a region, the less likely it covers a planar surface patch anyway.

- **Region eccentricity**
  Regions that have been extracted from the image but that are too *stretched* (i.e. one dimension being much longer than the other one) are rejected, as the intensity profile within such a region is again sensitive to discretization errors and has insufficient discriminative power. This decision is made on the basis of an *eccentricity value* derived from the eccentricity of the ellipse inscribed in the parallelogram.

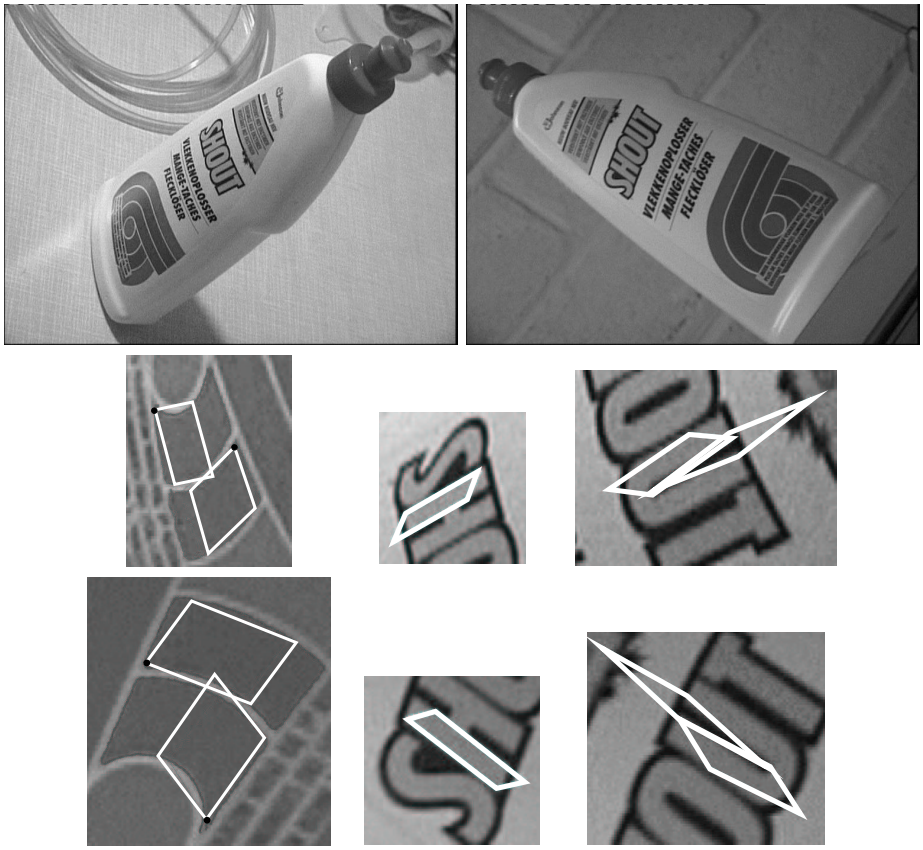


Figure 4.5: *Some affinely invariant regions based on corners and curved edges.*

For an overview of the other, less critical parameters and the default values assigned to them all, we refer to appendix C.

Example

Figure 4.5 shows some examples of affinely invariant regions extracted with the geometry-based method for curved edges. Although there is a large image distortion between the two images (geometrically as well as photometrically), the affinely invariant parallelograms (white) – which have been found for these images independently – clearly cover similar physical regions.

Figures 4.6 and 4.7 show a more worked out example. The parallelogram-shaped region found for the upper image has also been extracted in the lower image (full lines), together with another region (dashed lines). This second region went undetected in the first image, since the edge on which the detection is based (added in black in Figure 4.6) has been interrupted.

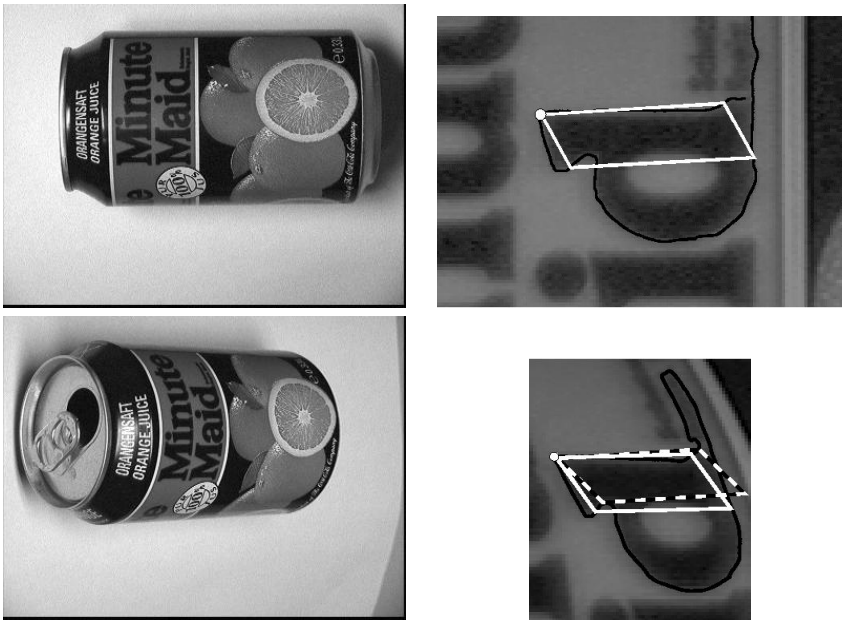


Figure 4.6: *Two images of the same object (left), with the affinely invariant regions found for one specific corner point (right).*

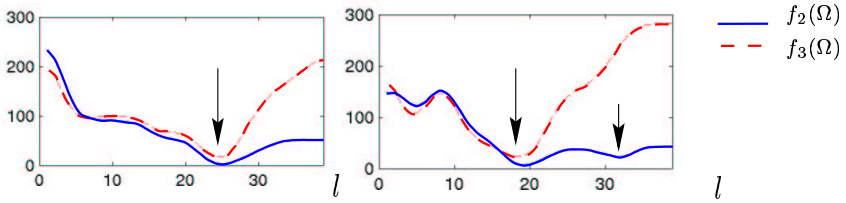


Figure 4.7: *Detection of local extrema of* $f_2(\Omega)$ *and* $f_3(\Omega)$

In Figure 4.7, the functions $f_2(\Omega)$ and $f_3(\Omega)$ used for the extraction of the regions in Figure 4.6 are shown. These have been computed based on the red band. Although the functions are not completely identical, the positions of their local extrema correspond (except at small values for l where noise and image discretization errors have a larger effect due to the small scale). As becomes clear from these graphs, the first region actually has been detected twice, once for $f_2(\Omega)$ and once again for $f_3(\Omega)$.

Note that the affinely invariant regions found with this method are *not* centered around the anchor point. A centered alternative is the parallelogram that has the non-centered parallelogram as one quadrant. Nevertheless, we prefer the non-centered regions, since experiments have shown that restricting

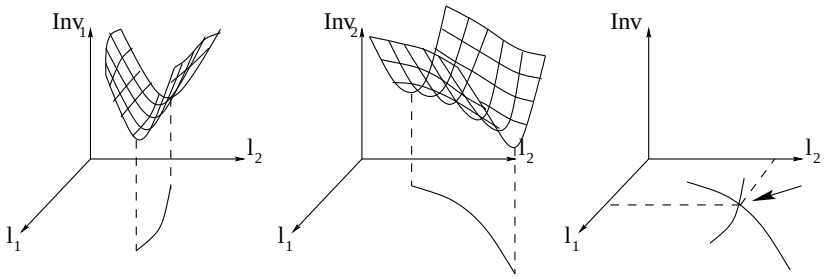


Figure 4.8: *For the straight edges case, the intersection of the "valleys" of two different functions is used instead of a local extremum.*

the region to only one quadrant (delineated by the edges) makes the assumption of planarity much more realistic. This is due to the fact that the anchor points we start from are corners, often lying near the border of an object.

4.3.2 Case 2: Straight edges

In the case of straight edges, the method described above cannot be applied, since $l = 0$ along the entire edge. However, since straight edges occur quite often, we cannot simply neglect this case.

A straightforward extension of the previous technique would then be to search for local extrema in a 2D search-space with the two parameters s_1 and s_2 for the two edges as coordinates, instead of a 1D search-space over l.

However, the functions $f_2(\Omega)$ and $f_3(\Omega)$ we used for the curved-edges case, do not show clear, well-defined extrema in the 2D case. Rather, we have some *valleys* of low values, (corresponding to cases where the center of gravity lies on or close to the diagonal). Instead of taking the inaccurate local extrema, we combine the two functions, and take the *intersection* of the two valleys, as shown in Figure 4.8.

Although this method seems rather ad-hoc and not a very principled one, the regions we obtain prove to be much more stable than those based on a 2D local extremum.

Implementation details

The straight edges case involves the evaluation of $f_2(\Omega)$ and $f_3(\Omega)$ on a 2D grid instead of a 1D grid as in the curved edges case. This 2D grid can be computed efficiently, in an incremental way, which is not possible in the curved edges case. To this end, we first *deskew* the image, based on the extracted straight edges. More precisely, we normalize the image to compensate for the skew by applying a coordinate transformation that makes the angle between the two edges equal to 90 degrees (see Figure 4.9).

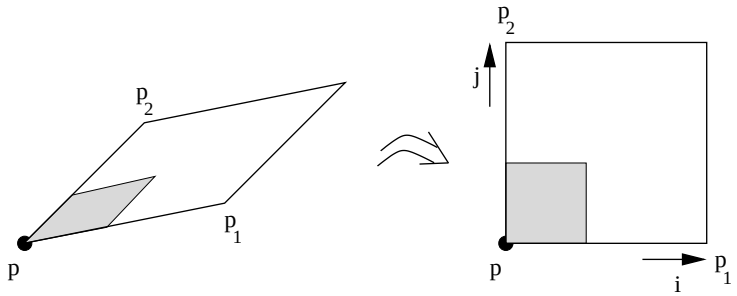


Figure 4.9: *Based on the straight edges found, the image can be 'deskewed' (i.e. normalized as to compensate for the skew), allowing an efficient computation of the moments.*

This allows to compute the moments involved in the computation of $f_2(\Omega)$ and $f_3(\Omega)$ in an incremental way as follows:

$$\begin{aligned} M_{pq}^{abc}(\Omega_{ij}) &= M_{pq}^{abc}(\Omega_{(i-1)j}) + M_{pq}^{abc}(\Omega_{i(j-1)}) - M_{pq}^{abc}(\Omega_{(i-1)(j-1)}) \\ &\quad + i^p j^q [R(i,j)]^a [G(i,j)]^b [B(i,j)]^c \end{aligned} \tag{4.8}$$

with Ω_{ij} the rectangular region extending from the origin of the deskewed region up to (i,j).

Computing the valley intersections is done in a rather straightforward way. First, the valleys of the two functions are located, by looking for local minima in horizontal and/or vertical direction. If a local minimum in at least one direction can be found for both functions, they are said to intersect at the corresponding position, and an affinely invariant region is created.

The special case where the two valleys (almost) coincide is detected and rejected, as this will result in many intersections close to one another. In order to limit the risk of such coinciding valleys, one of the two functions is computed on a smaller subregion Ω' which further reduces the correlation between the two functions.

Parameters

Apart from the extra line fitting parameters, the parameters to be specified for the straight edges case are similar to those of the curved edges case. For the line fitting, an implementation available in the TargetJr software distribution has been used [TJr], which has as main parameters the minimum line length in pixels (default value 15) and the maximum deviation from a line in degrees (default value 3).

For a detailed overview of all the parameters involved we refer to appendix C.

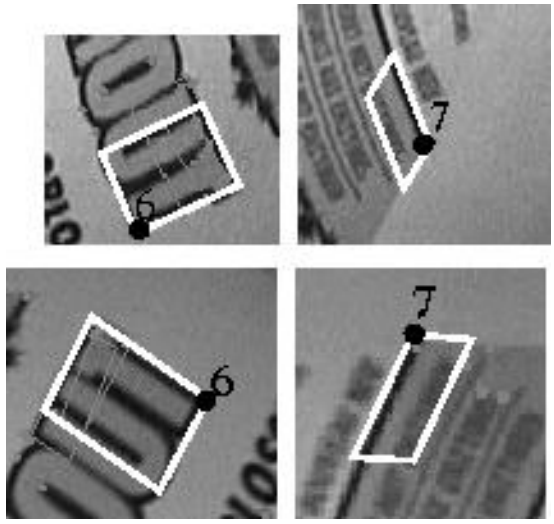


Figure 4.10: *Some local invariant regions extracted from the images shown in Figure 4.5 based on the geometry-based method for straight edges.*

Example

As a first example, some affinely invariant regions extracted with the method for straight edges found in the images of Figure 4.5 are shown in Figure 4.10.

A more detailed example of invariant regions found with the geometry-based method for straight edges can be found in Figures 4.11-4.14. Figure 4.11 shows the original images (left) and the affinely invariant regions found for a specific corner (right). Note the substantial amount of foreshortening in the lower image. Nevertheless, the same regions as found in the upper image have also been extracted in the lower image (full lines), together with a few other regions (dashed lines). The reason why these additional regions have not been found in the upper image is twofold. First of all, the change in scale is too large. The corresponding regions in the upper image would surpass the specified maximum region size. Secondly, some of these regions are not planar, covering parts of two sides of the box, one of which is invisible in the upper image.

Let us have a closer look at how exactly these regions have been found. Figure 4.12 shows some of the preprocessing steps involved in the region extraction: Harris corner detection (left), Canny edge detection (middle) and line fitting (right). Note the robustness to changes in viewpoint (or repeatability) of the Harris corner points: in spite of the high degree of foreshortening between both images, most corners found in one of the images have been found in the other image as well. The Canny edges perform a lot worse in this context. Certainly the smaller edges are not stable at all. The longer, more outspoken edges perform better, although they are not always connected in the same way.

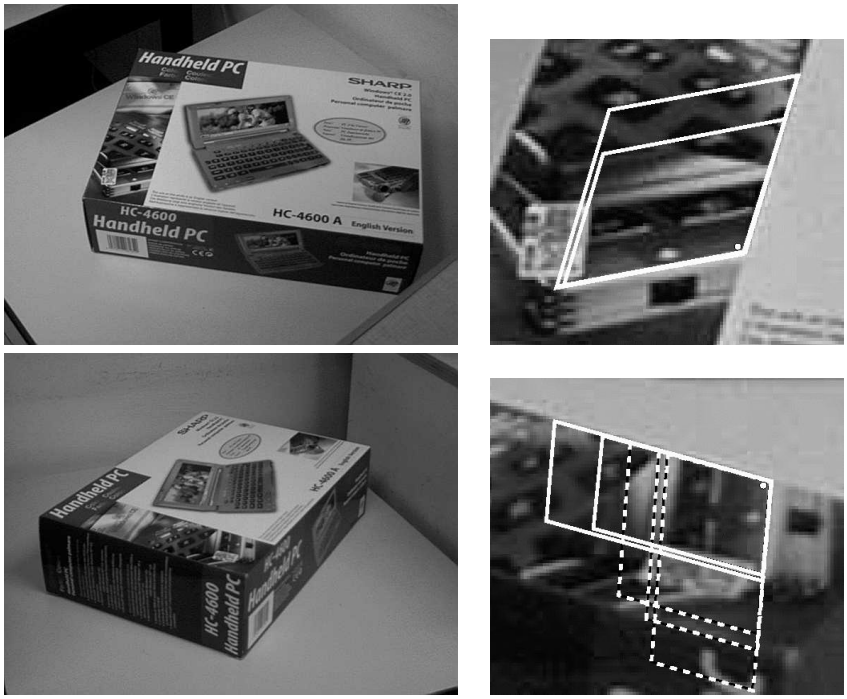


Figure 4.11: *Two images of the same object (left), with the affinely invariant regions found for one specific corner point (right).*

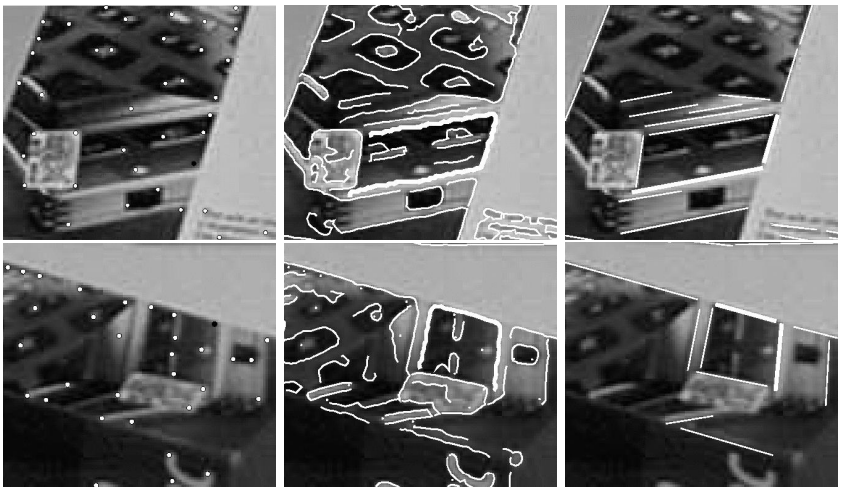


Figure 4.12: *Some preprocessing steps: detecting corner points (left), edges (middle) and lines (right).*

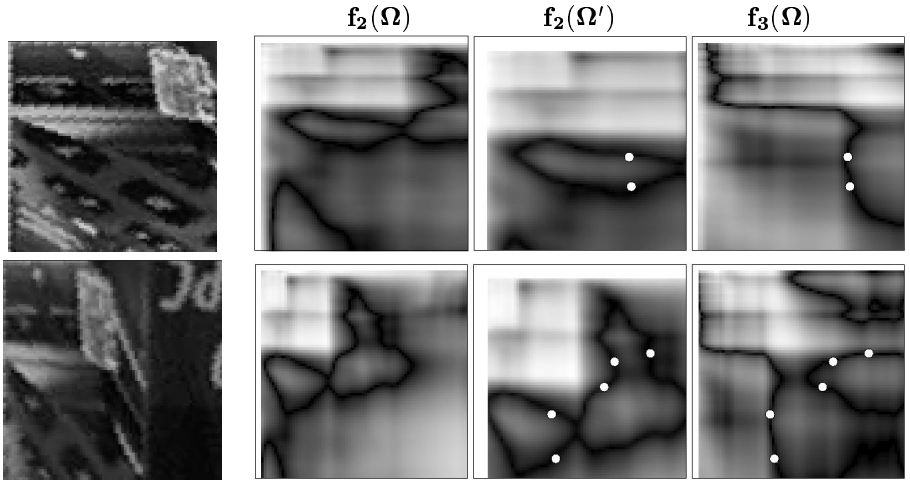


Figure 4.13: *Result of deskewing the neighborhood based on the detected lines (left), evaluating the functions $f_i(\Omega)$ over the regions (the darker a pixel the lower the value for $f_i(\Omega)$) and taking the intersections (white points) of the 'valleys' of these functions.*

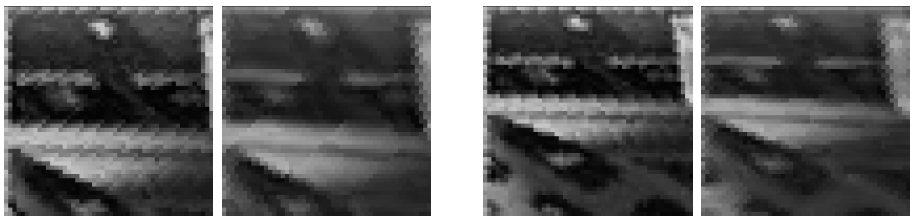


Figure 4.14: *After deskewing, the regions found for the first image (shown at the left of each pair) match two of the regions found for the second image (shown at the right of each pair).*

As the line fitting only uses the longer edges, they are again rather robust to viewpoint changes. In this figure, the edges and lines used for the extraction of the regions of Figure 4.11 are highlighted.

Based on the two straight edges intersecting near the selected corner point, the part of the image in between the two lines can be deskewed, as explained above. The deskewed images (up to maximum region size) are shown at the left of Figure 4.13. The skew has been compensated for, but there is still a substantial difference in scale between both images (both along the x and y axes). Based on these deskewed images, $f_2(\Omega)$ and $f_3(\Omega)$ can be computed in an efficient way using Equation 4.8. These functions are shown at the right of Figure 4.13. The images should be interpreted as follows: the darker the pixel at position (i, j), the lower the corresponding function evaluated over the rectangular region Ω_{ij} extending from the origin up to (i, j). The 'valleys of low values' clearly stand out as darker lines in these images. For taking the intersection, we do not use $f_2(\Omega)$ directly, but rather $f_2(\Omega')$, where Ω' is a subregion of Ω. This boils down to rescaling the image representing $f_2(\Omega)$ before taking the intersections (see Figure 4.13). The highlighted points indicate the positions of the valley intersections found.

As a final verification, the corresponding regions after deskewing and rescaling to a square of fixed size are shown in Figure 4.14. In theory, these images should be identical except for the photometric changes. In practice, the differences are very small. They are mainly caused by discretization errors and noise.

4.4 Intensity-based region extraction

A drawback of the method described in the previous section is that the edges it relies on are often a source of errors. Edges that were found in one image may be undetected, interrupted or connected in a different way in the second image. This section presents an alternative method for extracting invariant regions, that is directly based on the analysis of image intensity, without extraction of features such as edges or corners, which we therefore coined *intensity-based* region extraction. It turns out to complement the previous method very well, in that invariant regions are typically found at different locations in the image.

Instead of starting from corner points, this method uses local extrema in intensity as anchor points (cfr. section 4.2.2). Given such a local extremum, the intensity function along rays emanating from the extremum is studied, as shown in Figure 4.15. The following function is then evaluated along each ray:

$$f_I(t) = \frac{abs(|I(t) - I_0|)}{max(\frac{\int_0^t abs(|I(t)-I_0|)dt}{t}, d)} \tag{4.9}$$

with t the Euclidean arclength along the ray, $I(t)$ the intensity at position t, I_0 the intensity extremum and d a small number which has been added to prevent a division by zero. The point for which this function reaches an extremum

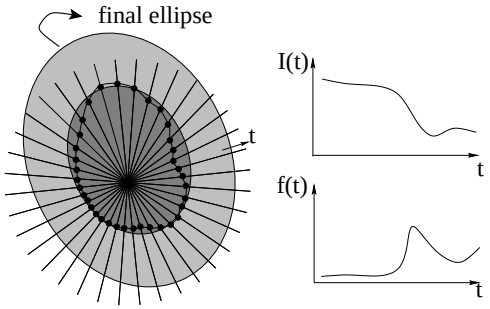


Figure 4.15: *The intensity along 'rays' emanating from a local extremum are studied. The point on each ray for which a function f(t) reaches an extremum is selected. Linking these points together yields an affinely invariant region, to which an ellipse is fitted using moments.*

is invariant under the affine geometric and photometric transformations we consider (given the ray). Typically, such extrema occur at positions where the intensity suddenly increases or decreases dramatically compared to the intensity changes we encountered on the line up to that point. Again, $f_I(t)$ is in itself already invariant. Nevertheless, we still select the points where this function reaches an extremum, as this is probably more robust then simply selecting points where the function reaches a specific value. For a proof of this invariance, we refer to appendix A.

Note that in theory, leaving out the denominator in the expression for $f_I(t)$ would yield a simpler function which still has invariant positions for its local extrema. In practice, however, this simpler function does not give good results since its local extrema are often too shallow, resulting in inaccurate positions along the rays and hence inaccurate regions. With the denominator added, on the other hand, the local extrema are mostly localized more accurately.

Next, all points corresponding to extrema of $f_I(t)$ along rays originating from the same local extremum are linked to enclose an (affinely invariant) region (see Figure 4.15). This often irregularly-shaped region is then replaced by an elliptic region having the same shape moments up to the second order. This ellipse-fitting is affinely invariant as well. Note that the resulting region is not centered around the original anchor point (the intensity extremum). In fact, the whole procedure is quite robust to the inaccurate localization of this point. In most cases, small changes in its position will have no effect on the resulting region if the intensity profile is indeed showing a shallow extremum.

Finally, we double the size of the elliptic region found. This leads to a higher discriminative power for the regions, due to a more diversified texture pattern within the region and hence facilitates the subsequent matching process, at the cost of a higher risk of non-planarity due to the less local character of the regions. This last step, which is not needed from a purely theoretical point of view, was found to drastically improve the matching results in a number of

experiments.

Implementation Details

Problems may arise when more than one local extremum can be found along the ray. Instead of choosing the global extremum, which would make the method sensitive to scale changes altering the search area along a ray, we select an extremum by imposing a *continuity constraint* as follows: in case of multiple extrema, we simply select the extremum closest to the extrema found along the neighboring rays.

Internally, the intensity-based, elliptic regions are represented in a similar way as the geometry-based regions, i.e. based on three points. The first point is the center of the ellipse. The two other points $\mathtt{p}_1$ and $\mathtt{p}_2$ are chosen in such a way that the affine transformation that maps the center of the ellipse to the origin and $\mathtt{p}_1$ and $\mathtt{p}_2$ to $(0,1)$ and $(1,0)$ respectively transforms the ellipse into the unit circle [3].

Parameters

The main parameters for the intensity-based region extraction are again the minimum and maximum region size and the region eccentricity threshold, which are all identical to the equivalent parameters for the geometry-based region extraction, and the number of 'rays' used, which is typically set to 50.

Again, we refer to appendix C for their default values.

Example

Figure 4.16 again shows some details of the two very different views shown in Figure 4.5, with some corresponding invariant regions that were extracted using the intensity-based method (white) and the linked points on which the region extraction is based (black).

In Figure 4.17, the intensity and function evaluation along two sample rays are shown. The lines superimposed on the images (upper row) represent the rays. These have been selected by hand, so they probably do not match perfectly. Each time, the function evaluations for the white ray are shown on the left below the corresponding image, while the function evaluations for the black ray are shown on the right. For comparison, we also added the evaluation of the numerator of $f_I(t)$. As already mentioned earlier, leaving out the denominator gives a simpler expression, whose local extrema are in theory also invariant, but seem to be less stable in practice.

[3] Note that this representation is not unique. One elliptic region can be represented by an infinity of point triples (corresponding to rotated versions of the unit circle).

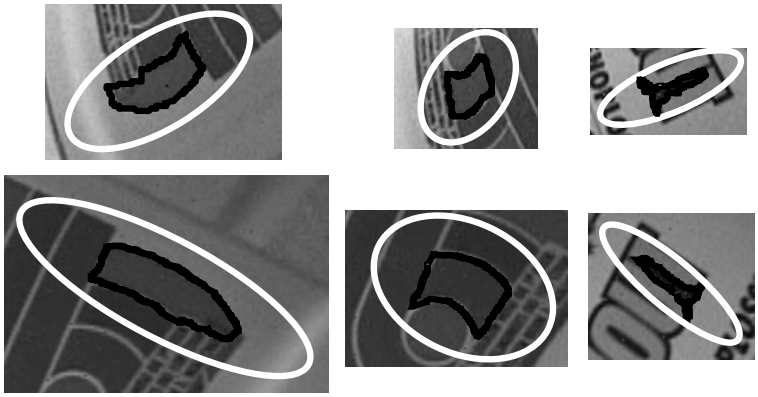


Figure 4.16: *Affinely invariant regions based on intensities only (white) and the linked points used to extract them (black).*

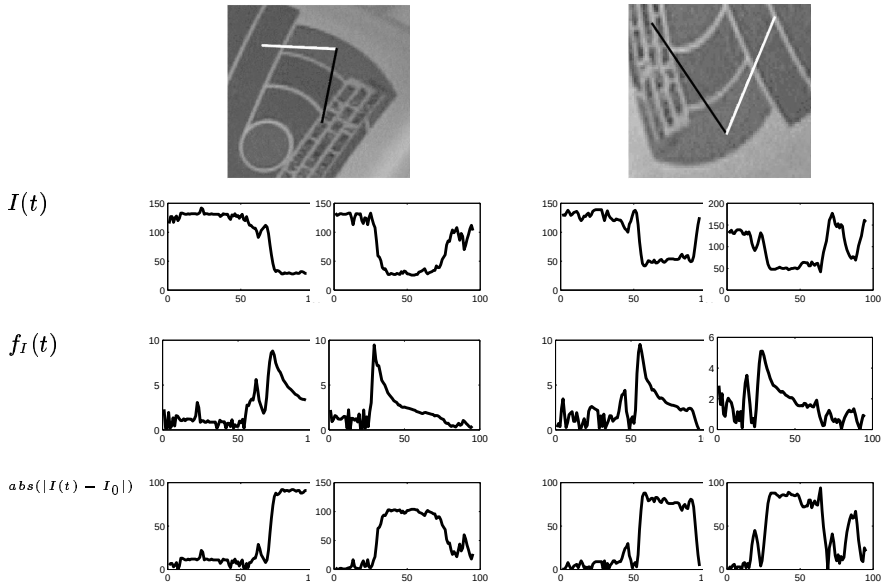


Figure 4.17: *Evaluation of $I(t)$, $f_I(t)$ and the numerator of $f_I(t)$ along two rays (evaluations along the white ray on the left side below each image, evaluations along the black ray on the right side below each image).*

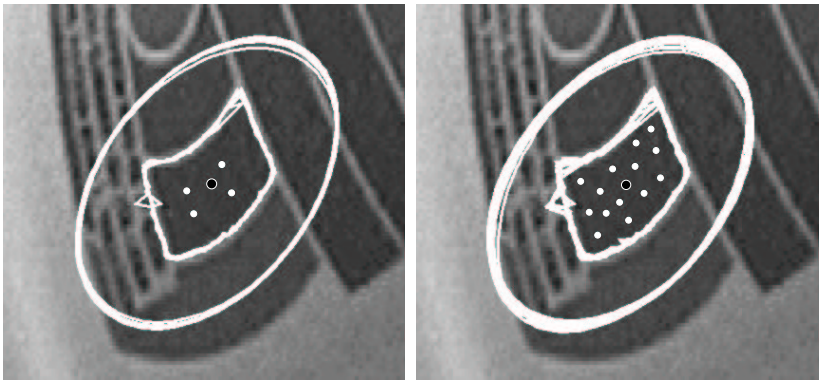


Figure 4.18: *Robustness of the region extraction to the inaccurate localization of the intensity extrema. Deviations in the position of the anchor point only have a minor impact on the resulting affinely invariant region.*

Robustness to inaccurately localized intensity extrema

As already mentioned in section 4.2, the major drawback of using intensity extrema as anchor points is the inaccurate localization of such points. This is caused by the fact that the intensity extrema are often rather flat, combined with the effect of the isotropic Gaussian smoothing for noise-reduction prior to the detection of local extrema, which is not affinely invariant. With this small experiment, we demonstrate that the region extraction as described above is sufficiently robust to this inaccurate localization of intensity extrema. To this end, we repeat the region extraction process starting from different anchor points lying close to the intensity extremum and having similar intensity values. As can be seen from Figure 4.18, moving around the anchor point has only a small effect on the resulting elliptic region found. The black point near the center of the figure is the original intensity extremum, while the white points are the anchor points used for the region extraction. To highlight the source of the deviations, we also added the linked extrema found along the rays. On the left part of Figure 4.18, only small deviations to the position of the anchor point are tested. Such deviations seem to have almost no effect on the final elliptic region. On the right part of the same figure, also larger deviations are tested. Now, the deviations on the final elliptic regions are a bit larger as well, but still acceptable.

So in general, the region extraction is quite robust to the inaccurate localization of the intensity extrema. However, the above no longer holds if the area enclosed by the linked extrema found along the rays is non-convex (as for instance in the example shown on the right in Figure 4.16).

4.5 Discussion

Two different methods for the extraction of local, affinely invariant regions have been proposed, a geometry-based method exploiting the corners and edges in the image and an intensity-based method.

Note that the regions extracted with our methods are typically larger than the ones used in most other object recognition systems based on local features (e.g. [SM97, Low99, HCVC00]). This is needed in order to obtain affine invariance in a robust way. Moreover, a trade-off has to be made between locality and discriminative power, certainly in case of such high level of invariance. For instance, we doubled the size of the elliptic, intensity-based regions, as otherwise it would be much harder to distinguish the individual regions.

Comparison

When comparing the geometry-based method with the intensity-based method, the following differences can be found:

- **Stability**
  Which method gives the most stable regions largely depends on the image content (e.g. whether it contains clearly outspoken edges or not). Overall, from our experiments we have the impression that the intensity-based method performs slightly better in this respect, although it is hard to prove this, as the result is so image-dependent.

- **Number of regions found**
  The geometry-based region extraction method can detect several regions of different sizes for a single corner point, corresponding to several local extrema or valley intersections. This introduces some kind of scale concept. The intensity-based method, on the other hand, gives at most one region for each intensity extremum (or none, if no local extremum is found along a ray). As a result, there are typically many more geometry-based regions in an image than intensity-based regions.

- **Efficiency/Computation time**
  Neither of the two region extraction methods is really fast. On a SGI O2 workstation, the geometry-based method typically takes 800 msec for the straight edges case and 140 msec for the curved edges case. The intensity-based method is even slower, with approximately 4 seconds for a single intensity extremum (if a region is found, otherwise, it is faster). Moreover, the detection of the intensity extrema takes more time than the Harris corner detection as well. To process the images of Figure 4.11, the geometry-based method needs about 4 minutes for around 300 corners, extracting about 1000 regions, while the intensity-based method spends about 4.5 minutes examining around 200 intensity extrema, resulting in around 50 regions. However, it should be noted that efficiency

was not really an issue during the implementation of these methods. The speed of several steps in the region computations can considerably be increased. Moreover, the process can easily be spread over several processors if needed, using different processors for different corner points or intensity extrema.

Robustness

The question remains how *robust* the proposed region extraction methods actually are. Several sources of errors can be identified, that make that a region found in one view of an object can go undetected in another view. Typically, only a certain percentage of the regions found in the first view will be found in the second view as well.

First of all, some of the basic assumptions on which we based our model of affine geometric and linear photometric transformations (three scalefactors and three offsets) may be violated. The following cases can be distinguished:

- The region covers a non-planar surface patch.
- The planar surface patch corresponding to a region found in one image is partly or completely occluded in the other image.
- There is too much perspective deformation, even on a local scale, in one of the images, due to the camera being too close to the object.
- There is some specular reflection on part of the object in one of the two images, or, more generally, there are non-linear changes in intensities.
- The change in scale between both images is too large, such that the corresponding region in the other image is too large or too small (compared to the minimum and maximum region size). A similar problem occurs in case of too much skew in the affine transformation, causing the region in one image being rejected because it is too eccentric.
  Also, a large change in scale can cause structures to appear at one scale that have gone undetected in the other image.
- Finally, noise and discretization errors have not been taken into account in our model. Certainly on a local scale, these can have a tremendous effect. We have tried to stay robust to these by working with moments wherever possible.

The effect of noise and discretization, combined with the effect of non-linearities in the photometric changes should not be underestimated. This is highlighted in Figure 4.19. It shows the normalized red bands of some deskewed regions as they were extracted from several images of the same object (also shown in Figure 4.11). Clearly, these regions 'match', i.e. they correspond to the same physical part of the object. The geometric deformations have been compensated for by deskewing the regions. The photometric deformations have been compensated for by rescaling the intensities to the interval $[0 - 255]$. According to our model, these images should be identical, which is clearly not



Figure 4.19: *Noise, discretization errors and non-linear intensity changes can cause large deviations from the modeled transformations.*

the case. Noise, discretization errors and non-linear intensity changes are at the basis of the differences found between those images. Nevertheless, in this case we were able to extract more or less the same region from the different images. However, in other cases, the deviations from our model may spoil the region extraction completely.

Apart from these general failures, each of the two extraction methods we proposed has its own weaknesses. The Achilles' heel of the geometry-based region extraction method is its dependence on the correct detection of the geometric entities. Harris corner detection and Canny edge detection are among the best feature extractors currently available. Nevertheless, they are *not* affinely invariant. Hence some corners found in one image are missing in another image of the same object. Edges are sometimes missing as well, or interrupted or connected in a different way, resulting in different regions being extracted. Also the line fitting is not infallible. For instance, the result depends on the scale as the threshold for the minimum line length is fixed. And even if all this works fine, the detection of local extrema or the valley intersection can still give erroneous results due to noise, etc.

The intensity-based region extraction is not perfect either. If the extrema are not sufficiently outspoken, they can be overwhelmed by noise. Also, from time to time the wrong decision is taken in case of several local extrema along a ray. Finally, problems can arise due to the inaccurate localization of the intensity extrema when the area enclosed by the linked extrema found along the rays is non-convex.

From the above, it becomes clear that individual regions are not always reliable. Given a *single* corner or local intensity extremum, it cannot be guaranteed that similar region(s) are extracted in two different views of an object. For the examples given in this chapter, it worked fine (even though we sometimes found more regions in one image than in the other), but we could include examples of failures due to each of the reasons mentioned above as well. As a consequence, missing matches as well as false matches are quite common. Nevertheless, considering *all* corners or intensity extrema in two views of an object, there will be both stable and unstable ones. A robust matching algorithm, as explained in the next chapter, can then match the stable ones while ignoring the unstable ones by testing the global consistency, and hence find a sufficient number of corresponding regions between the two views for most applications,

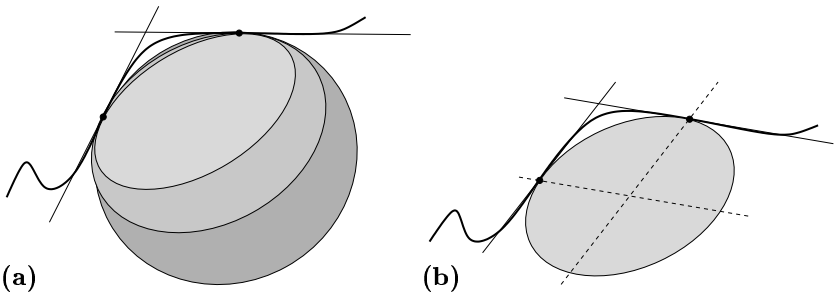


Figure 4.20: *Alternative construction of affinely invariant elliptic region starting from two inflection points on a curve.*

even in case of large viewpoint changes.

Alternative construction methods

We do not claim the two region extraction methods proposed in this chapter to be the Holy Grail of local affinely invariant regions. They are just two examples of how one could extract such regions, which proved sufficiently stable and reliable (provided one does not rely on a single region). As such, they demonstrate the feasibility of local, affinely invariant region extraction. However, other constructions are possible as well and can be added to our opportunistic system to further improve its performance. In this section, two alternative region extraction methods are sketched. They start from a single curve and its inflection points.

As a first example of an alternative region extraction method, one could start from a curve and consider ellipses that are tangent to the curve in two neighboring inflection points (see Figure 4.20a). A one-dimensional family of ellipses fulfills these conditions. Based on the color or intensity profile within the ellipses, one or maybe a few ellipses can be selected to form a new kind of elliptic affinely invariant regions (e.g. by looking for local extrema of some function evaluated over the elliptic regions). Alternatively, one can draw lines through each of the inflection points parallel to the tangent line in the other inflection point. The intersection of these two lines then determines the center of an ellipse, that is tangent to the curve in the two inflection points (see Figure 4.20b).

A drawback of both variations is that they depend on the position of the inflection points, which is not that accurate. Also, the second approach is not guaranteed to result in a *local* region either. Still, the biggest problem with this technique is again its dependence on the accurate detection of the edges in the image. Moreover, a larger segment of the curve will typically be needed, making it less local and hence even more sensitive to edge interruptions etc.

Another alternative construction method is illustrated in Figure 4.21. It starts from a single inflection point, and uses two floating points $\mathtt{p}_1$ and $\mathtt{p}_1{}'$

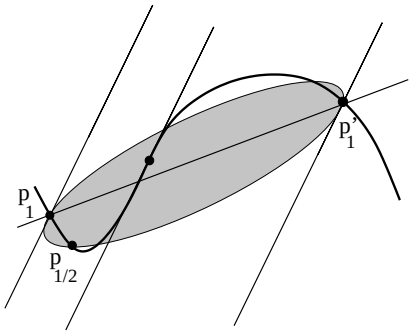


Figure 4.21: *Alternative construction of affinely invariant elliptic region starting from a single inflection point on a curve.*

along the edge (one in each direction). These points start from the inflection point and are linked through a relative affinely invariant parameter l, similar to the first part of the geometry-based method for curved edges. We also define a third point $\mathtt{p}_{1/2}$ along the curve in one direction, where the relative affinely invariant parameter reaches half the value at $\mathtt{p}_1$ and $\mathtt{p}_1{}'$. Then, we can construct an ellipse that goes through all three points, and whose tangent lines in $\mathtt{p}_1$ and $\mathtt{p}_1{}'$ are parallel to the tangent line to the curve in the inflection point. Based on the intensity profile within the elliptic region, one or a few elliptic regions are selected as a function of l.

The advantage of this method is that it does not depend on the position of the inflection point along the edge [4]. This will probably result in more stable regions. A disadvantage of this method, on the other hand, is that the final region lies on both sides of the edge, increasing the risk of non-planarity. Also, the regions will often be too eccentric to be useful.

4.6 Conclusion

As we do not believe in a single type of region that works for all possible images under all possible circumstances, we propose a more *opportunistic* system that exploits several types of regions simultaneously, depending on what's on offer. In this way, we want to arrive at a robust and flexible system that is applicable to a wide range of images. And indeed, we successfully applied the region extraction methods proposed in this chapter to images of indoor as well as outdoor scenes, images of paintings, buildings, and so on (see Chapter 6-10).

Two different region extraction methods have been proposed. The first one is coined the *geometry-based method*, as it firmly relies on the geometric entities present in the image. It starts from a corner and its nearby edges. Two

[4] The relative affinely invariant parameter l is zero as long as the curve follows the tangent line, so the positions of $\mathtt{p}_1$ and $\mathtt{p}_1{}'$ are independent of the position of the inflection point.

points move away from the corner in both directions along the edges. Their relative speed is coupled through the equality of relative affinely invariant parameters. At each position, the two points together with the corner define a parallelogram-shaped region. The points stop at positions where simple photometric quantities of the texture covered by the parallelogram go through an extremum, thereby selecting a specific parallelogram. The evaluated photometric quantities have been chosen in such a way that the whole procedure is invariant under the considered geometric and photometric transformations.

The second region extraction method does not rely on the accurate extraction of geometric entities from the image, but directly exploits the raw image data. That's why we have coined it the *intensity-based method*. It starts from local intensity extrema, and examines the intensity profile along rays emanating from such an extremum to select a specific point on each ray. An elliptic region is then fitted to the connection of all these points in an invariant way.

Although we usually work with color images (applying the region extraction methods to each of the three spectral bands), the methods for region extraction proposed in this chapter can be used for grayscale images as well, as they work on a single band only.

As shown by the examples in this chapter, and further on in the experimental validation, these two methods are capable of extracting corresponding regions in two different views of the same object or scene, even in case of large viewpoint changes. It should be noted though that the correct extraction of *individual* regions is not always reliable. The proposed methods are vulnerable to non-planarity, perspective deformations, non-linear intensity changes, discretization errors, image noise, occlusions and so on. All these effects can cause the failure of the extraction for a specific region. However, when considering *all* regions in an image or belonging to an object, a sufficient amount of regions will normally survive the transformations. With the robust matching algorithm proposed in the next chapter, we can match the stable regions, while ignoring the unstable ones.

Chapter 5

Matching local, affinely invariant regions

This chapter is organized as follows. First, the description of the color or intensity pattern within a region using affine moment invariants is explained in section 5.1. Next, different methods to compare the intensity patterns of two regions are proposed in section 5.2. In section 5.3, the different feature vectors proposed in section 5.1 are compared, based on their discriminative power. In section 5.4, the process of 'matching' regions between two images is described, i.e. for each region in the first image, how to find the corresponding region (if any) in the second image and vice versa. Section 5.5 discusses some semi-local constraints that can be used to reject false matches and hence to increase the robustness of the system. Section 5.6 concludes the chapter.

5.1 Invariant region description

Now that we have methods at hand for delineating local, affinely invariant regions, we can finally extract local, affine invariants, computed based on the same support independent of the viewpoint. Such invariants are indispensable for an efficient matching of local features under large changes in viewpoint. As in the region extraction step, we consider invariance both under affine geometric changes and linear photometric changes with different offsets and different scale factors for each of the three color bands.

Each region is characterized by a feature vector of moment invariants. The moments we use are *generalized color moments* [MMVG99a] (see section 3.2.2). They are based on powers of image coordinates and intensities of different color channels:

$$M_{pq}^{abc} = \iint_{\Omega} x^p y^q [R(x,y)]^a [G(x,y)]^b [B(x,y)]^c \, dxdy \tag{5.1}$$

In fact, they implicitly characterize the shape, the intensity and the color distribution of the region pattern in a uniform manner. We have opted for this kind of invariants, as they use all information available on a local scale (geometry, texture *and* color) to the fullest extent and in a robust way.

Several different sets of invariants are proposed here (see Tables 5.1-5.3). Which one performs best will be tested experimentally in section 5.3, after we have explained the matching procedure in section 5.2.

A generic affinely invariant feature vector

The first feature vector of affine moment invariants we propose is also the most generic one, as it can be used for *any* kind of affinely invariant regions, in contrast to the other feature vectors, which exploit some special properties of specific types of regions.

This feature vector is composed of 18 moment invariants. These are invariant functions of moments up to the first order and second degree (i.e. moments that use up to second order powers of intensities (R, G, B) and first order powers of (x, y) coordinates). In [MMVG99b], it has been proven that the 18 invariants form a basis for all invariants under the considered group of geometric and photometric transformations involving this kind of moments. For an overview of the 18 invariants, see Table 5.1. The notations used in this table are similar to those used in [MMVG99b].

The reason why these expressions are so complex is that they also take the photometric transformations fully into account (i.e. both scaling and offset for each spectral band). As a result, it is hard to give a physical interpretation. Nevertheless, we can say that the first three invariants are related to the area defined by the center of the region, the center of gravity weighted with one color-band and the center of gravity weighted with the same color-band squared (relative to the total area of the region, and with an extra factor to compensate for the effect of the intensity offset to the positions of the centers of gravity). The next three invariants ($inv[4]$ to $inv[6]$) are related to the correlation between two different color-bands. $inv[7]$ to $inv[9]$ are similar to the first three invariants, but now related to the relative positions of the center of gravity weighted with two different color-bands. $inv[10]$ to $inv[18]$ are based on similar combinations of several centers of gravity (e.g. R vs RG for $inv[10]$, or R vs G^2 for $inv[16]$). However, with centers of gravity weighted with combinations of different color-bands, it becomes more and more complex to compensate for the illumination changes, hence the growing complexity of the expressions.

Normalized feature vectors

The feature vector of affine moment invariants described in the previous paragraph is generally applicable to any type of affinely invariant regions. The advantage is that this allows to treat all regions in the same way. Also, this

makes it easy to add new types of regions to our system.

Table 5.1: *Moment invariants used for comparing the patterns within an invariant region.*

$inv[1] = S_{12}^{R} = \dfrac{\left\{ \begin{array}{l} M_{10}^{200} M_{01}^{100} M_{00}^{000} - M_{10}^{200} M_{00}^{100} M_{01}^{000} - M_{01}^{200} M_{10}^{100} M_{00}^{000} \\ +M_{01}^{200} M_{00}^{100} M_{10}^{000} + M_{00}^{200} M_{10}^{100} M_{01}^{000} - M_{00}^{200} M_{01}^{100} M_{10}^{000} \end{array} \right\}^2}{(M_{00}^{000})^2 \left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2 \right]^3}$ $inv[2] = S_{12}^{G} \qquad (similar)$ $inv[3] = S_{12}^{B} \qquad (similar)$
$inv[4] = D_{02}^{RG} = \dfrac{\left[M_{00}^{110} M_{00}^{000} - M_{00}^{100} M_{00}^{010} \right]^2}{\left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2 \right] \left[M_{00}^{020} M_{00}^{000} - (M_{00}^{010})^2 \right]}$ $inv[5] = D_{02}^{GB} \qquad (similar)$ $inv[6] = D_{02}^{BR} \qquad (similar)$
$inv[7] = D_{12}^{1RG} = \dfrac{\left\{ \begin{array}{l} M_{10}^{100} M_{01}^{010} M_{00}^{000} - M_{10}^{100} M_{00}^{010} M_{01}^{000} - M_{01}^{100} M_{10}^{010} M_{00}^{000} \\ +M_{01}^{100} M_{00}^{010} M_{10}^{000} + M_{00}^{100} M_{10}^{010} M_{01}^{000} - M_{00}^{100} M_{01}^{010} M_{10}^{000} \end{array} \right\}^2}{(M_{00}^{000})^4 \left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2 \right] \left[M_{00}^{020} M_{00}^{000} - (M_{00}^{010})^2 \right]}$ $inv[8] = D_{12}^{1GB} \qquad (similar)$ $inv[9] = D_{12}^{1BR} \qquad (similar)$
$inv[10] = D_{12}^{2RG} = \dfrac{\left\{ \begin{array}{l} (M_{00}^{000})^2 M_{10}^{100} M_{01}^{020} - M_{00}^{000} M_{10}^{100} M_{01}^{000} M_{00}^{020} \\ -2M_{00}^{000} M_{01}^{010} M_{00}^{010} M_{10}^{100} + 2M_{01}^{000} (M_{00}^{010})^2 M_{10}^{100} \\ -M_{00}^{000} M_{10}^{000} M_{00}^{100} M_{01}^{020} + 2M_{10}^{000} M_{00}^{010} M_{00}^{100} M_{01}^{010} \\ -(M_{00}^{000})^2 M_{01}^{100} M_{10}^{020} + M_{00}^{000} M_{01}^{100} M_{10}^{000} M_{00}^{020} \\ +2M_{00}^{000} M_{10}^{010} M_{00}^{010} M_{01}^{100} - 2M_{10}^{000} (M_{00}^{010})^2 M_{01}^{100} \\ +M_{00}^{000} M_{01}^{000} M_{00}^{100} M_{10}^{020} - 2M_{10}^{010} M_{00}^{100} M_{01}^{000} M_{00}^{010} \end{array} \right\}}{(M_{00}^{000})^4 \left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2 \right] \left[M_{00}^{020} M_{00}^{000} - (M_{00}^{010})^2 \right]^2}$ $inv[11] = D_{12}^{2GB} \qquad (similar)$ $inv[12] = D_{12}^{2BR} \qquad (similar)$
$inv[13] = D_{12}^{3RG} = \dfrac{\left\{ \begin{array}{l} (M_{00}^{000})^2 M_{10}^{010} M_{01}^{200} - M_{00}^{000} M_{10}^{010} M_{01}^{000} M_{00}^{200} \\ -2M_{00}^{000} M_{01}^{100} M_{00}^{100} M_{10}^{010} + 2M_{01}^{000} (M_{00}^{100})^2 M_{10}^{010} \\ -M_{00}^{000} M_{10}^{000} M_{00}^{010} M_{01}^{200} + 2M_{10}^{000} M_{00}^{100} M_{00}^{010} M_{01}^{100} \\ -(M_{00}^{000})^2 M_{01}^{010} M_{10}^{200} + M_{00}^{000} M_{01}^{010} M_{10}^{000} M_{00}^{200} \\ +2M_{00}^{000} M_{10}^{100} M_{00}^{100} M_{01}^{010} - 2M_{10}^{000} (M_{00}^{100})^2 M_{010}^{010} \\ +M_{00}^{000} M_{01}^{000} M_{00}^{010} M_{10}^{200} - 2M_{10}^{100} M_{00}^{100} M_{01}^{000} M_{00}^{010} \end{array} \right\}^2}{(M_{00}^{000})^4 \left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2 \right]^2 \left[M_{00}^{020} M_{00}^{000} - (M_{00}^{010})^2 \right]}$ $inv[14] = D_{12}^{3GB} \qquad (similar)$ $inv[15] = D_{12}^{3BR} \qquad (similar)$

Table 5.1: *Moment invariants used for comparing the patterns within an invariant region (continued).*

$$inv[16] = D_{12}^{4RG} = \frac{\left\{ \begin{array}{l} (M_{00}^{000})^2 M_{10}^{100} M_{01}^{110} - M_{00}^{000} M_{10}^{100} M_{01}^{000} M_{00}^{110} \\ -M_{00}^{000} M_{10}^{100} M_{00}^{100} M_{01}^{010} + M_{10}^{100} M_{00}^{100} M_{01}^{000} M_{00}^{010} \\ -M_{00}^{000} M_{10}^{000} M_{00}^{100} M_{01}^{110} + M_{10}^{000} (M_{00}^{100})^2 M_{01}^{010} \\ -M_{10}^{000} M_{00}^{100} M_{00}^{010} M_{01}^{100} - (M_{00}^{000})^2 M_{01}^{100} M_{10}^{110} \\ +M_{00}^{000} M_{01}^{100} M_{10}^{000} M_{00}^{110} + M_{00}^{000} M_{01}^{100} M_{00}^{100} M_{10}^{010} \\ +M_{00}^{000} M_{01}^{000} M_{00}^{100} M_{10}^{110} - M_{01}^{000} (M_{00}^{100})^2 M_{10}^{010} \end{array} \right\}^2}{(M_{00}^{000})^4 \left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2\right]^2 \left[M_{00}^{020} M_{00}^{000} - (M_{00}^{010})^2\right]}$$

$inv[17] = D_{12}^{4GB}$ $(similar)$

$inv[18] = D_{12}^{4BR}$ $(similar)$

However, sometimes knowledge about the region extraction can be used to derive easier invariant expressions, using lower order moments and hence resulting in more stable feature descriptions. This can be achieved by *normalizing* the region against some transformations.

Normalization for the geometry-based regions In case of regions extracted with the geometry-based method, we know that the resulting region is parallelogram-shaped. The skew and scale changes can be removed simply by applying an additional affine transformation that maps the parallelogram onto a square reference region of fixed size. Moreover, we also know which corner of the parallelogram corresponds to the original Harris corner point. Based on this information, we can compensate for the rotation. The only geometric deformation left is a possible 'switching' of the two axes (due to the edges being taken in a different order in the two images). However, since we ordered the edges in a clockwise fashion, this only happens when the object itself is mirrored. For most applications we can safely neglect this case[1].

In this way, the affine transformations have completely been compensated for. In fact, the normalization described above corresponds to using the points $\mathtt{p}$, $\mathtt{p}_1$ and $\mathtt{p}_2$ as an affine basis, and describing the color or intensity profile with respect to this basis.

We also normalize against the photometric changes. However, this cannot be achieved by exploiting some extra knowledge about the region extraction. Instead, this normalization is directly based on the intensity profile itself, by replacing each intensity value I by $I' = aI + b$ with a and b such that the average intensity is 128 and with a spread on the intensities of 50.

For an overview of the invariants used in this case, we refer to Table 5.2. As all transformations have been compensated for through normalization, the

[1]For grouping purposes, where symmetric features may be mirrored variations of one another, switching of the axes can occur. In that case, a variation of the invariant feature vector proposed here should be used.

Table 5.2: *Moment invariants used for comparing the patterns within a parallelogram-shaped invariant region after normalization of the region to a reference square.*

$inv[1] = \frac{M_{00}^{110}}{M_{00}^{000}}$	$inv[2] = \frac{M_{00}^{011}}{M_{00}^{000}}$	$inv[3] = \frac{M_{00}^{101}}{M_{00}^{000}}$
$inv[4] = \frac{M_{10}^{100}}{M_{00}^{100}}$	$inv[5] = \frac{M_{10}^{010}}{M_{00}^{010}}$	$inv[6] = \frac{M_{10}^{001}}{M_{00}^{001}}$
$inv[7] = \frac{M_{01}^{100}}{M_{00}^{100}}$	$inv[8] = \frac{M_{01}^{010}}{M_{00}^{010}}$	$inv[9] = \frac{M_{01}^{001}}{M_{00}^{001}}$
$inv[10] = \frac{M_{11}^{100}}{M_{00}^{100}}$	$inv[11] = \frac{M_{11}^{010}}{M_{00}^{010}}$	$inv[12] = \frac{M_{11}^{001}}{M_{00}^{001}}$
$inv[13] = \frac{M_{20}^{100}}{M_{00}^{100}}$	$inv[14] = \frac{M_{20}^{010}}{M_{00}^{010}}$	$inv[15] = \frac{M_{20}^{001}}{M_{00}^{001}}$
$inv[16] = \frac{M_{02}^{100}}{M_{00}^{100}}$	$inv[17] = \frac{M_{02}^{010}}{M_{00}^{010}}$	$inv[18] = \frac{M_{02}^{001}}{M_{00}^{001}}$

invariants become very simple. In fact, *any* measurement in the normalized reference region can be used as an invariant. The reason why we still stick to moments is that they are more robust to noise.

Note that the invariants we propose here do not necessarily form a basis. Actually, they were selected rather ad-hoc based on their physical interpretation. $inv[1]$ to $inv[3]$ are related to the correlation between two color-bands. $inv[4]$ to $inv[6]$ and $inv[7]$ to $inv[9]$ are the x- and y-coordinates respectively of the centers of gravity weighted with one color-band, while $inv[10]$ to $inv[18]$ are some combinations of higher order moments.

Normalization for the intensity-based regions When using the intensity-based region extraction method, we obtain elliptic regions. A similar normalization as for the geometry-based, parallelogram-shaped regions can be applied, this time using a circular reference region instead of a square. However, there remains one degree of freedom to be determined, namely the orientation of the color or intensity profile within the circular reference region. So the invariants used in this case should still be rotation-invariant. Again, normalization against illumination changes has been used as well.

Table 5.3 summarizes the invariants for this case. Only 9 invariants are given here, although it is possible to find more by combining several color-bands and using second order moments.

$inv[1]$ to $inv[3]$ are identical to the corresponding invariants for the geometry-based, normalized regions (representing the correlation between several color bands). $inv[4]$ to $inv[6]$ are the distances between the center of gravity of the region weighted with one color band and the center point of the region, while $inv[7]$ to $inv[9]$ correspond to the weighted average squared distances to the

Table 5.3: *Moment invariants used for comparing the patterns within an elliptic invariant region after normalization to a reference circular region.*

$$inv[1] = \frac{M_{00}^{110}}{M_{00}^{000}} \qquad inv[2] = \frac{M_{00}^{011}}{M_{00}^{000}} \qquad inv[3] = \frac{M_{00}^{101}}{M_{00}^{000}}$$

$$inv[4] = \frac{\sqrt{\begin{array}{l}(M_{10}^{100}M_{00}^{000})^2 + (M_{10}^{000}M_{00}^{100})^2 + (M_{01}^{100}M_{00}^{000})^2 + (M_{01}^{000}M_{00}^{100})^2 \\ -2M_{10}^{100}M_{00}^{000}M_{10}^{000}M_{00}^{100} - 2M_{01}^{100}M_{00}^{000}M_{01}^{000}M_{00}^{100}\end{array}}}{M_{00}^{100}M_{00}^{000}}$$

$inv[5]$ $(similar)$

$inv[6]$ $(similar)$

$$inv[7] = \sqrt{\frac{\begin{array}{l}M_{20}^{100}(M_{00}^{000})^2 - 2M_{10}^{000}M_{10}^{100}M_{00}^{000} + M_{10}^{000}M_{10}^{000}M_{00}^{100} \\ +M_{02}^{100}(M_{00}^{000})^2 - 2M_{01}^{000}M_{01}^{100}M_{00}^{000} + M_{01}^{000}M_{01}^{000}M_{00}^{100}\end{array}}{M_{00}^{100}(M_{00}^{000})^2}}$$

$inv[8]$ $(similar)$

$inv[9]$ $(similar)$

center of the region (x_m, y_m), i.e. $inv[7] = \sqrt{\frac{\int\int [(x-x_m)^2+(y-ym)^2]Rdxdy}{\int Rdxdy}}$.

A feature vector for grayscale images

Whenever color information is available, it is advised to use it for the invariant region description, as it provides additional information, making it easier to discriminate different regions. However, in some cases, only grayscale images are available. In that case, only the single-band invariants can be used. From Table 5.1, it becomes clear that only one such invariant exists when using only moments of second order and first degree and without normalization. This is clearly insufficient to distinguish regions. So normalization should be used, whenever possible. Alternatively, extra invariants can be derived based on moments of higher order or degree, or the region can be divided in two subregions (e.g. a darker and a brighter half), which again allows for the computation of additional, lower order invariants (see [MMVG98]).

5.2 Comparing two regions

In this section, we discuss several methods that can be used to compare the color or intensity patterns within two regions. The first method is based on the Mahalanobis distance between the corresponding feature vectors of affine moment invariants (section 5.2.1). The second method compares regions by computing the crosscorrelation of the intensity patterns (section 5.2.2). Finally, section 5.2.3 gives some measures based on maximum likelihood.

5.2.1 Comparing two feature vectors

In theory, two feature vectors computed over the same regions but under different viewing conditions should be identical, as they are composed of *invariants*. In practice, they will never be completely equal, due to noise, discretization errors or misalignments between the two regions. Another source of errors are deviations from the model used to describe the deformations between the two images. Remember that we assumed affine geometric deformations and linear photometric deformations. These models won't be adequate if there is a perspective deformation in the image, if the surface is not completely planar, if it is (partially) occluded or if there is a strong specular reflection. Specular reflections in particular can turn the intensity profiles of the two regions (and hence also the invariant descriptions) completely different, making it impossible to match them. However, as this occurs only occasionally and mostly on a local scale, we can safely neglect this case. The other deviations are usually small, such that the two feature vectors will still be relatively close to one another.

Mahalanobis distance To find out whether two feature vectors are close enough to one another (i.e. find out whether the two feature vectors might be representing the same region), we need a good distance measure. Clearly, a Euclidean distance is not good enough in this case, since the variation or spread on the different elements of the feature vector is different. The variance on one invariant may be several orders of magnitude larger than the variance on another invariant. At the same time, several invariants may be correlated as well.

The *Mahalanobis distance* is a better distance measure in this case, as it correctly takes into account this different variability of the elements of the feature vector. It is computed as follows:

$$d_M(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{\Lambda}^{-1} (\mathbf{x}_1 - \mathbf{x}_2)} \tag{5.2}$$

with $\mathbf{x}_1$ and $\mathbf{x}_2$ the feature vectors that are being compared and $\mathbf{\Lambda}$ the *covariance matrix* (i.e. Λ_{ii} represents the *spread* on the ith element of the feature vector and $\Lambda_{ij}/\sqrt{\Lambda_{ii}\Lambda_{jj}}$ $(i \neq j)$ the *correlation* between the ith and jth component).

The importance of good statistics Unfortunately, it is not straightforward to obtain a good estimate for the covariance matrix $\mathbf{\Lambda}$, as the variation of the invariants strongly depends on the degree of viewpoint and illumination change, as well as on the intensity profile of the region. Due to the first dependency, the statistics should be computed based on images covering a wide range of possible viewpoints, illumination conditions etc. Moreover, due to the dependency on the intensity profile, the statistics are different for each region, so they should be recomputed over and over again. This clearly is unfeasible from a practical point of view. Instead, the covariance matrix must be approximated, based on some reasonable assumptions.

In fact, each region can be considered a separate class. Often, it is then assumed that the intra-class statistics can well be approximated based on the inter-class statistics (i.e. the overall distribution of the features of all regions in an image). The underlying rationale is that the latter already give a good idea of the dynamic range of each of the features and the correlation between different features.

However, this approximation is far from being optimal. In fact, in most cases the variability of a single feature over different viewing conditions is hardly related to the overall variability of that feature. Some features may be very stable to image noise or changing viewing conditions, resulting in a small intra-class variability, while they still vary a lot between different regions due to the different intensity profiles. Other features, such as those using higher order moments, are far less discriminative, as they are much more sensitive to image noise, misalignment of the region or imperfections of the model, irrespective of the overall variability of the feature. Actually, a good feature is a feature with a high *discriminative power*, i.e. the combination of a high variability between different regions with a low variability over different viewing conditions. This corresponds to applying a *Fischer analysis* instead of a principal component analysis [Fis38], which combines high inter-class variances with low intra-class variances by applying two consecutive coordinate transformations.

This is based on the less strict *common covariance matrix assumption*, i.e. it is assumed that all classes (regions) share the same intra-class statistics (independent of the color or intensity profile of the regions). However, these statistics can be different from the overall, inter-class statistics. It has been shown that, if the number of training samples is small, this assumption leads to higher classification accuracy, even if the covariance matrix of each class greatly differs [Fri89].

We select a few representative regions and extract the corresponding feature vectors under varying viewing conditions (viewpoint, illumination, scale, etc.). After subtracting the mean of the appropriate class, all samples contribute to the computation of one common covariance matrix, which is used in all experiments to minimize the intra-class variations. Maximizing the inter-class variations can be done for each experiment separately, tuned to the specific statistics of the regions in a specific image or database.

Indexing techniques The nice thing about comparing regions based on the Mahalanobis distance between invariant feature vectors is that it can be implemented in an efficient way, using *indexing* or *hashing* techniques. To find a similar feature vector, we do not have to perform a sequential search (i.e. check the regions one after the other, each time computing the Mahalanobis distance or any other measure for comparison, then accept or reject). Instead, we can use (part of) the feature vector as a key or index for table lookup, which directly yields a list of similar feature vectors [Wol90]. In this way, finding the most similar region becomes less dependent on the total number of matching candidates.

5.2.2 Crosscorrelation-based comparison of regions

Apart from comparing the invariant descriptions of two regions, one can also directly compare corresponding pixel intensities in the two regions by computing the crosscorrelation. Crosscorrelation-based comparison of local regions is the most widely used technique to find correspondences between two images of the same object or scene. The definition of the *normalized crosscorrelation* is given by

$$\mathcal{C} = \frac{\sum_i \sum_j \left[I(x+i,y+j) - \widehat{I}\right]\left[I'(x'+i,y'+j) - \widehat{I'}\right]}{\sqrt{\sum_i \sum_j \left[I(x+i,y+j) - \widehat{I}\right]^2 \sum_i \sum_j \left[I'(x'+i,y'+j) - \widehat{I'}\right]^2}} \tag{5.3}$$

with (x, y) and (x', y') corresponding pixels in the first respectively the second image, $I(x, y)$ and $I'(x', y')$ the respective intensities and $\widehat{I}$ and $\widehat{I'}$ the average intensities.

$\mathcal{C}$ is a measure for the similarity between the two intensity patterns. Its value always lies between -1 and 1. It is equal to ± 1 if and only if the two intensity patterns are related by a linear relation. The lower the absolute value of the normalized crosscorrelation, the less correlated the two patterns are. In the rest of this text, we simply use the term 'crosscorrelation' when referring to the normalized crosscorrelation.

However, the crosscorrelation as such is not robust to the image distortions caused by changing viewpoint. It assumes that the pixel corresponding to $(x + i, y + j)$ is located at $(x' + i, y' + j)$, which is only the case under a pure two-dimensional translation. To overcome this problem, we compute the crosscorrelation after transforming the original regions to unit square or circular reference regions (squares for the parallelogram-shaped regions and circular regions for the elliptic regions). In this way, the geometric deformations are compensated for, except for the rotational component of the circular regions, which is found by maximizing the crosscorrelation.

The crosscorrelation is a more powerful measure than the Mahalanobis distance. However, it cannot be used as efficiently as the Mahalanobis distance. In order to find a corresponding region in an image, *all* regions in the image have to be tested one after the other, such that the needed computation time is linear in the number of matching candidates.

5.2.3 Comparison based on maximum likelihood measures

A crosscorrelation-based comparison of two regions is based on the assumption of *linear* changes in image intensities. Indeed, the crosscorrelation directly measures the linear dependency of the two intensity patterns. Similarly, the invariants used for the comparison based on the Mahalanobis distance also assume a linear model for the intensity changes. However, as shown by the examples in Figure 4.19 in the previous chapter, this linear model is not always

exact. Invariants based on more complex, non-linear models for the photometric deformations can be developed. However, it is difficult to find a good, accurate mathematical model for this phenomenon. In this section, we propose some other measures based on *maximum likelihood*, that can correctly deal with non-linear and unmodeled intensity transformations. However, we have not yet implemented these measures in our system. We just give them here as a suggestion for further improvement.

Using *mutual information*, it is possible to measure the similarity between two regions without imposing a specific model for the relationship between corresponding intensity values. This is a technique introduced by Viola [VW97], based on ideas from information theory. The mutual information is measured as follows:

$$\mathcal{I}(I,J) = \sum_i \sum_j \log\left(\frac{p(i,j)}{p(i)\,p(j)}\right) \tag{5.4}$$

with $p(i)$ the probability of intensity i in image I, $p(j)$ the probability of intensity j in image J and $p(i,j)$ the joint probability of intensity i in image I and intensity j in image J. This measure expresses the idea that the color or intensity profile of one region should be predictable to a high degree based on the color or intensity profile of the other region. The pure existence of a (statistical) relationship between the intensity values in the first region and the corresponding intensity values in the other region suffices to obtain high scores with this measure. There is no need to model the exact form of the relationship.

Alternatively, the *correlation ratio* can be used [RocAl98, RMA99] as well.

$$\eta(I,J) = \frac{Var\{E(j|i)\}}{Var\{j\}} \tag{5.5}$$

with $E(J|I)$ the expected value of the intensity in image J given the intensity of the corresponding pixel in image I and $Var\{...\}$ the variance of the measure between brackets. Instead of a statistical relationship, this measure tests for the existence of a *functional* relationship, i.e. each color or intensity value in the first region is mapped to a single color or intensity value in the other region (or, more precisely, a distribution around a single color or intensity value). The difference with mutual information is that the correlation ratio is based on the variance instead of the entropy.

Although the relation between corresponding pixel intensities in two different regions corresponding to the same physical part of an object is not exactly linear (see Figure 4.19), it can most of the time be assumed that pixels with identical colors in one image also have identical or similar colors in the other image. Hence, the correlation ratio will probably give more accurate results than the mutual information measure. Moreover, it can be proved that the value obtained for the correlation ratio is always bigger or equal to the cross-correlation [RMA99].

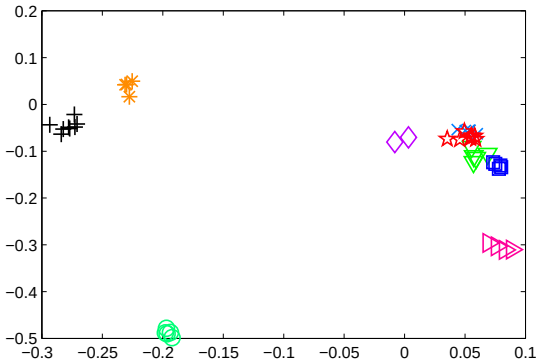


Figure 5.1: *Fischer analysis for the feature vector of Table 5.1*

5.3 Comparison of the different feature vectors

From the discussion in section 5.2.1, we know that a *Fischer analysis* should be applied if one wants to compare the discriminative power of the different feature vectors given in Table 5.1-5.3.

For this experiment, two sets of representative regions are extracted from a few series of images of the same objects, taken from different viewpoints, illumination conditions, scale, etc. Based on the *first* set of regions, the covariance matrix of the intra-class statistics for each type of feature vector is estimated (after a manual classification i.e. matching of the regions). Next, the *second* set of regions is used to test the discriminative power of the different feature vectors. Based on the covariance matrix for the intra-class statistics derived using the first set of regions, a first coordinate transformation is performed, that should minimize the variability between different feature vectors representing the same region. Then, a second principal component analysis is applied, to maximize the inter-class variability.

The results are summarized in Figures 5.1-5.3 for the feature vectors of Table 5.1-5.3 respectively. In the figures, the feature vectors projected onto the two largest principal components are shown. Each time, points with the same symbol represent feature vectors computed for corresponding regions in different images. (i.e. regions representing the same physical part of an object). Due to the first PCA (based on the intra-class statistics), feature vectors belonging to the same class (i.e. corresponding regions in different images, represented by the same symbol) should be close to one another. Due to the second PCA, feature vectors for different regions should be far away from one another. As a result, the features should form well separated clusters, allowing easy classification. The better separated the clusters, the higher the discriminative power of the corresponding feature vector.

From this analysis, we can draw the following conclusions:

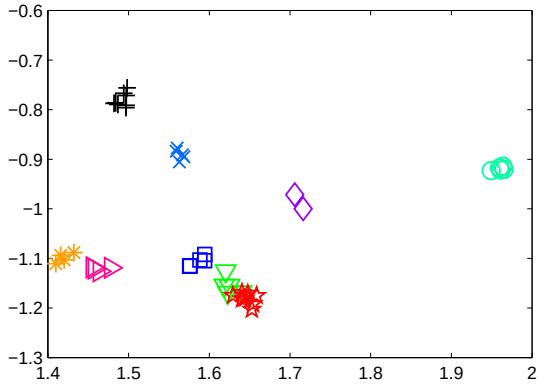


Figure 5.2: *Fischer analysis for the feature vector of Table 5.2*

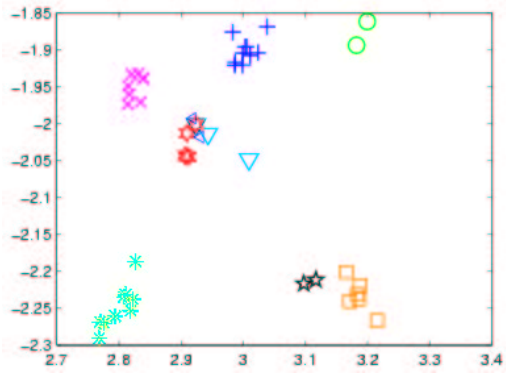


Figure 5.3: *Fischer analysis for the feature vector of Table 5.3*

- The *normalized* feature vectors (given in Table 5.2-5.3, and shown in Figures 5.2 and 5.3) outperform the generic feature vector of Table 5.1 (shown in Figure 5.1), as the clusters for these feature vectors are better separated from one another. So whenever possible, it is advized to use the normalized feature vectors that exploit the extra information about a specific type of region. That is why, in the experiments in the rest of this work, we always work with the normalized feature vectors from now on (unless explicitly mentioned otherwise).
- Several times, the same invariant is computed based on different color-bands. Such invariants are usually quite correlated, although the combination of all three invariants still has more discriminative power than each invariant individually.
- A Fischer analysis definitely gives better results than a principal component analysis.

5.4 Matching regions between two images

Now that we know which feature vectors perform best, and which methods are available to compare regions, it is about time to develop a strategy for bringing into correspondence the regions found in two different images of the same object or scene.

We start with extracting the affinely invariant regions in each image. Next, each region in the first image is matched to the region in the second image yielding the smallest possible Mahalanobis distance (which should also be below a predefined threshold d). However, for each anchor point at most one match is allowed, to prevent contradictions. If more than one region belonging to one and the same anchor point have been matched, only the most similar match is retained (i.e. the one with the smallest Mahalanobis distance). This can be done in an efficient way, using indexing techniques as explained in section 5.2.1. Next, all regions found in the second image are matched in a similar way to the regions found in the first image. Only a *mutual* match is accepted as a real correspondence between the two views.

Apart from the invariants listed in Table 5.1- 5.3, we added the *region type* as an additional invariant. This value refers to the method that has been used for the region extraction (intensity-based or geometry-based, straight edges or curved edges, extrema of which function, etc.). Only if the type of two regions corresponds, can they be matched.

Once corresponding regions have been found, the crosscorrelation between them (after normalization to a square or circular reference region) is computed as a final check before accepting the region correspondence. In case of elliptic regions, the correct rotation between the two circular reference regions is determined by maximizing the crosscorrelation. Again, simple thresholding is applied. If a match is rejected because of the crosscorrelation, the second

best match is tested (i.e. the match with the smallest but one Mahalanobis distance), and so on until at most ten matches have been tested.

Good matches are then further optimized, in the following way: keeping the region in one image fixed, small deviations on the parameters of the corresponding region in the other image are tried out, in order to maximize the crosscorrelation, using Levenberg-Marquardt non-linear optimisation. Usually, small changes in those parameters (e.g. changing the position of one of the corners of the parallelogram by one pixel) can already lead to substantial increases in the value of the crosscorrelation. If the parameters of the region drift away too far from their original value during this optimisation step, this may be a sign that it is actually an incorrect match, and the region correspondence is rejected. This step may be important if the matches are used in a subsequent step to determine the epipolar geometry, as this process is very sensitive to noise and hence needs accurate input. For other applications such as object recognition or content-based image retrieval, this step may not be needed. Further on, we will refer to this step as the *finetuning* of the regions.

Finally, some semi-local constraints are used to test the consistency of the matches found, which allows to reduce the number of false matches. How this is done will be discussed in more detail in section 5.5.

Parameters

A typical value for the threshold for the Mahalanobis distance between feature vectors is 2. Assuming that the covariance matrix has correctly been estimated and that the invariants have a Gaussian distribution, the probability of not being within this boundary although the regions have correctly been extracted from both images is smaller than 5 %.

The threshold for the crosscorrelation is initially set to 0.7. Then, the regions are finetuned to maximize the crosscorrelation, after which the value should at least be 0.8.

Example

Figure 5.4 shows the result of applying the above matching algorithm to two different views of the same object. At first, 24 matching candidates were found (based on the Mahalanobis distance and the crosscorrelation tests), 18 of which were actually correct (shown in white in Figure 5.4), 4 of which were caused by symmetries in the image (due to the same text and images on different sides of the box), and 2 matches were completely false (both symmetric and false matches shown in black in Figure 5.4). After checking the global consistency by applying the semi-local constraints, only one false match and two symmetric matches were left. Checking for a consistent epipolar geometry using RANSAC allowed to reject these remaining false and symmetric matches, leaving only the 18 correct matches.



Figure 5.4: *Example of matching regions between two images*

5.5 Semi-local constraints

Due to the wide range of geometric and photometric transformations allowed and the local character of the regions, false correspondences are inevitable. These can be caused by symmetries or repetitive structures in the image, or simply because the local region's distinctive power is insufficient.

Semi-local or *global* constraints offer a way out: by checking the consistency between combinations of local correspondences (assuming a rigid motion), false correspondences can be identified and rejected. Here, two new semi-local constraints are proposed that may be used to reject outliers. Both work on a combination of two region correspondences only, hence the amount of combinatorics needed is limited. The first one tests the geometric consistency, while the second one is a photometric constraint.

This is akin to the work of Carlsson [Car00], who has recently proposed a view compatibility constraint for five points in two views based on a scaled orthographic camera model. McReynolds and Lowe [MRL96] on the other hand developed a method for rigidity checking, working on six points in an iterative way, which is closely related to the equations for structure from motion. Finally, Sinclair *et al.* [SCR95] have derived a rigidity constraint based on two homographies, which we will discuss in more detail further on.

Checking the epipolar constraint

Probably the best known global constraint is the *epipolar constraint* (see Equation 3.13 and Figure 3.5). Checking for a consistent epipolar geometry is a widely applied method to reject false matches [TBM94, ZhaAl95]. Several techniques have been proposed based on robust statistics [Rou87]. Here, we incorporated the RANSAC (or RANdom SAmpling Consensus) approach [FB81] used by Torr *et al.* [TBM94, Tor95]. Combinations of seven points are selected randomly and used to compute a hypothesized epipolar geometry. Then, the support for this epipolar geometry is tested. If a sufficiently large number of the matches is found to be in agreement with the hypothesized epipolar geometry, the hypothesis is accepted. Otherwise, another sample of seven points is tried out, and the same process is repeated until a good hypothesis is found. In this way, all correspondences not conform with the found epipolar geometry can be rejected, while all the others have been proven to be consistent.

Although this method works fine in many applications, our experiments have shown that it may have difficulties in a typical wide baseline stereo setup, where false matches may outnumber the good ones and the total number of matches is limited. In that case, many of the randomly selected seven-point samples contain outliers, resulting in large computation times (each time rejecting the sample and trying out a new combination), or even erroneous results (a sample containing an outlier coincidentally yielding a reasonable amount of matches) That is why we have developed some extra, simple tests that can be used as a first, rudimentary filtering before checking the epipolar geometry.

A geometric constraint

Each match between two affinely invariant regions defines an affine transformation, which is, in turn, an approximation of the homography linking the projections of all points lying in the same plane. All possible plane-related homographies between two images span a four-dimensional subspace of the nine-dimensional space of 3×3 matrices [LF96, Sha96]. However, this does not mean that a combination of more than four different homographies (i.e. four region correspondences) is needed before one is able to derive a constraint. Due to the special structure of this four-dimensional subspace, a constraint can already be derived given just two different homographies. It is even possible to derive the epipolar geometry starting from just two homographies [PZ98b].

Suppose we have two homographies H_1 and H_2, belonging to planes Π_1 and Π_2 respectively. Combining them as $H_1^{-1}H_2$ yields a planar homology, whose eigen-analysis reveals one fixed point (the epipole) and one line of fixed points (the common line of the planes Π_1 and Π_2). This line of fixed points is used by Sinclair *et al.* [SCR95] to test whether two rigid planar motions are compatible. To this end, this common line is projected to the other image using H_1, and once again using H_2. If the two planes are indeed in rigid motion, the two resulting lines in the second image should coincide, which can easily be checked.

The geometric constraint we derive here is an algebraic distance. As it needs only a combination of two region correspondences and only requires the evaluation of the determinant of a 3×3 matrix, it can be applied quite fast. This makes it well suited for applications like ours, where a large number of consistency checks are performed on different combinations of region correspondences.

To check whether two correspondences found are geometrically consistent with one another, it suffices to check whether

$$det \begin{pmatrix} a_{23} - b_{23} & b_{13} - a_{13} & a_{13}b_{23} - b_{13}a_{23} \\ a_{22} - b_{22} & b_{12} - a_{12} & a_{12}b_{23} - b_{13}a_{22} + a_{13}b_{22} - b_{12}a_{23} \\ a_{21} - b_{21} & b_{11} - a_{11} & a_{11}b_{23} - b_{13}a_{21} + a_{13}b_{21} - b_{11}a_{23} \end{pmatrix} \leq \delta_g \quad (5.6)$$

with δ_g a predefined threshold and $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ the affine transformations mapping the region in the first image to the region in the second image, for the first and second match respectively. For the derivation of this semi-local constraint, we refer to appendix B.

Suppose we have N correspondences, each linking a different local region in image I to a similar region in image I' by N different affine transformations. Given a combination of two such correspondences, we first check if they are not coplanar (i.e. identical affine transformations). If not, the above consistency constraint can be checked. A specific region correspondence is considered incorrect if it is consistent with less than n_g other correspondences (with n_g typically 8). Each good correspondence should have at least n_g other consistent correspondences. This procedure may have to be repeated a number of times,

since rejecting a correspondence may cause other correspondences to have their number of consistent correspondences decreased below the threshold as well.

A photometric constraint

Apart from geometric constraints, photometric constraints may be derived as well. Although it is not necessarily true that the illumination conditions are constant over the entire image (due to shadows, multiple light sources, etc.), it is reasonable to assume that at least some parts of the images have similar illumination conditions as the region correspondence under scrutiny. So for each region correspondence, one should be able to find at least n_{ph} other region correspondences with a similar transform in the intensities (with n_{ph} typically 4).

First, the linear transformations (three offsets and three scalefactors) linking the intensities in both images are computed for each of the region correspondences using moments. Then, these transformations are compared. To be consistent, only an overall scale factor is allowed, to compensate for different orientations of the planes. If not enough consistent region correspondences can be found, the region correspondence is rejected. Again, a few iterations may be needed.

Example

Checking these constraints first before testing the epipolar constraint using RANSAC can considerably improve the results under the hard conditions of wide baseline stereo. As an example, consider the images shown at the top of Figure 5.5. Due to the repetition on the keyboard, there are many wrong matches at first, since more or less the same regions were found for most of the keys. As less than 50 percent of the matches are correct, directly applying RANSAC to find the epipolar geometry is very difficult. After having checked 2000 samples, our implementation of RANSAC bailed out, without having found the correct solution yet. Based on the semi-local constraints, 17 of these wrong matches were rejected. From the 26 matches left, 21 were correct, 3 were still caused by symmetries on the keyboard and 2 were completely false matches. All of these are shown superimposed on the images in the middle part of Figure 5.5. Checking for a consistent epipolar geometry using RANSAC based on these 26 matches, allowed to reject the five outliers and find the correct epipolar geometry after checking just 37 samples. The accuracy of the extracted epipolar geometry can be checked visually at the lower part of Figure 5.5.

5.6 Conclusions

Based on the methods for affinely invariant region extraction proposed in the previous chapter, it is now possible to derive invariants even on a local scale that can cope with the considered geometric and photometric deformations, as

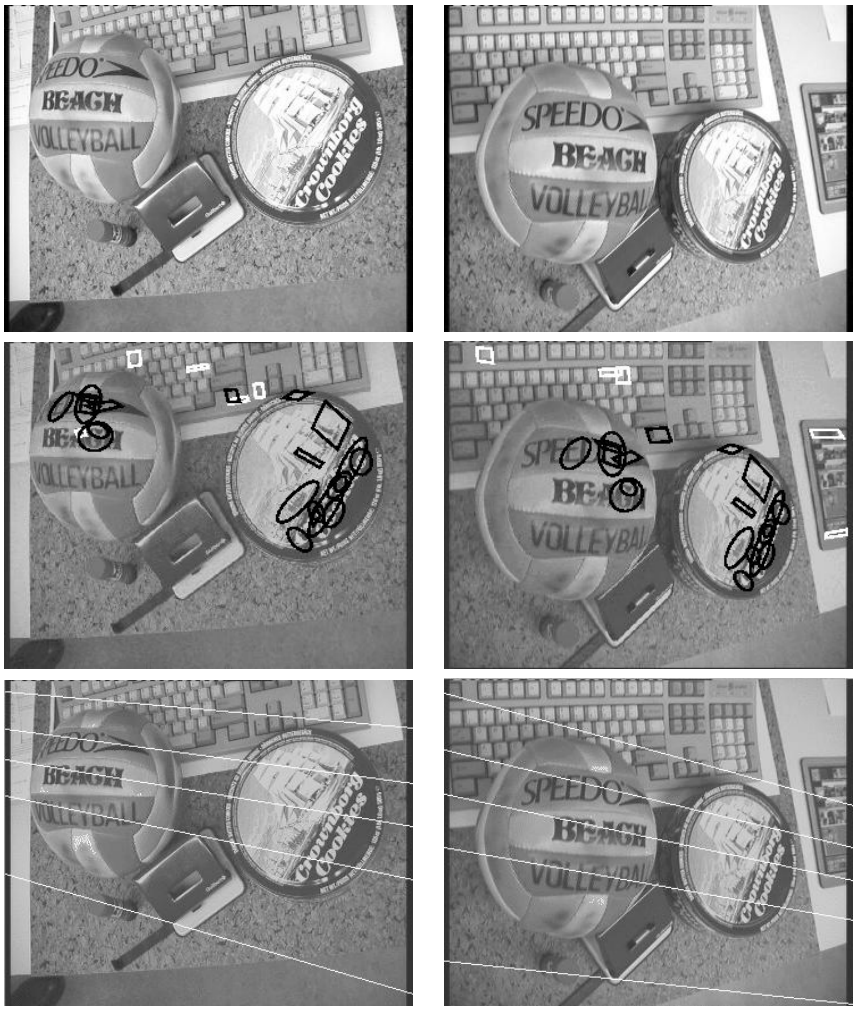


Figure 5.5: *Example of rejection of false matches using semi-local constraints*

they can be computed based on the same support independent of the viewpoint. In this chapter, several feature vectors of affine moment invariants have been proposed.

Based on these feature vectors, it is possible to find correspondences between the regions extracted in two different images of the same object or scene. However, since the regions are computed on such a local scale, there will often be some false matches, caused by symmetries in the image or simply due to the lack of sufficient discriminative power of the region's content. These can be rejected by testing the crosscorrelation between both regions, after normalization to a square or circular reference region, and by applying some simple semi-local constraints.

Chapter 6

Experimental validation

No man's knowledge can go beyond his experience.
John Locke.

In this chapter, the proposed affinely invariant region extraction methods and robust matching proposed in the previous chapters are experimentally validated. More precisely, we test how well the proposed methods survive all the variations that may be expected when using the system in real world applications. We test the robustness of the system to changes in viewpoint, scale, illumination, noise and image blur. We start with a short introduction (section 6.1), followed by the experimental results (section 6.2). Section 6.3 concludes the chapter.

6.1 Introduction

Before we start with the experimental results, we have to determine how we shall validate the quality of the region extraction and matching. As already mentioned in chapter 4.5, the region extraction is not very reliable if one focuses on a single anchor point. Due to the small scale of the regions and the large variations that may occur (noise, illumination changes, discretization errors, non-planarity, occlusions, etc.) it cannot be guaranteed that the same region is found in two different views of the same object or scene. Typically, only a certain percentage of the regions found will be common to the two views. Here, we consider two different measures to qualify the region extraction and matching: first, the absolute number of regions that have been matched correctly between two images, which should be as high as possible and at least above 8, such that the epipolar geometry can be computed using linear techniques; and second, the percentage of false matches, which should, of course, be as small as possible, preferably zero.

Concerning the false matches, we make a distinction between *symmetric* matches and *false* matches. The first class corresponds to those regions that

have not been matched to the region covering the same physical part of the object, but to another, similar region, that cannot be distinguished from the correct one due to a symmetry or repetitive structure in the image (e.g. another, identical window of the same building). It's the combination of these two measures that determines how good the system performs in practical situations. For instance, they determine the accuracy of the resulting epipolar geometry, the correct recognition of an object under varying viewing conditions or the performance of a database retrieval system based on the matching of local invariant regions. For object recognition specifically, the results of this chapter give an idea of how far the system can generalize from a single view, and hence allow to determine the number of reference images needed to model an object.

6.2 Experimental validation

6.2.1 Robustness to viewpoint changes

To quantitatively check the viewpoint invariance of our method, we took images of a drink can starting from head on and gradually increasing the viewing angle in steps of 10 degrees. The results of this experiment are summarized in Figure 6.1.

The images were taken with a digital camera mounted on the end-effector of a robot arm, moving in a circular fashion. As in almost all experiments in this chapter (except the experiment testing the robustness to illumination changes), the images are of size 576×768 pixels (i.e. PAL resolution).

For each image, the affinely invariant regions were extracted, and matched to the regions found in the 0 degrees reference image. Next, the regions were finetuned to optimize the crosscorrelation and filtered using the semi-local geometric and photometric constraints. Finally, we tested for a consistent epipolar geometry using RANSAC to automatically select the 'good' matches, and verified these matches visually, subdividing them into three different categories: correct, symmetric and false. The symmetric matches found in this experiment are mainly due to the text on the drink can which contains twice the letter 'M'. Moreover, these letters are exactly below one another, such that they lie more or less on the same epipolar line as well due to the chosen camera movement. So there is no way for the system to distinguish the regions on these two letters.

The horizontal line added to the figure indicates the lower threshold of 8 correct matches needed for computing the epipolar geometry in a simple (linear) way.

From Figure 6.1, one can see that for this example, the system can deal with changes in viewpoint up to more or less 60 degrees. Only correct and symmetric matches were left. For larger angles, the epipolar test could no longer be applied, as the number of matches was too low. It is mainly the change in scale due to the foreshortening of the object that causes problems, in combination with more and more specular reflection causing non-linearities in the intensity changes.

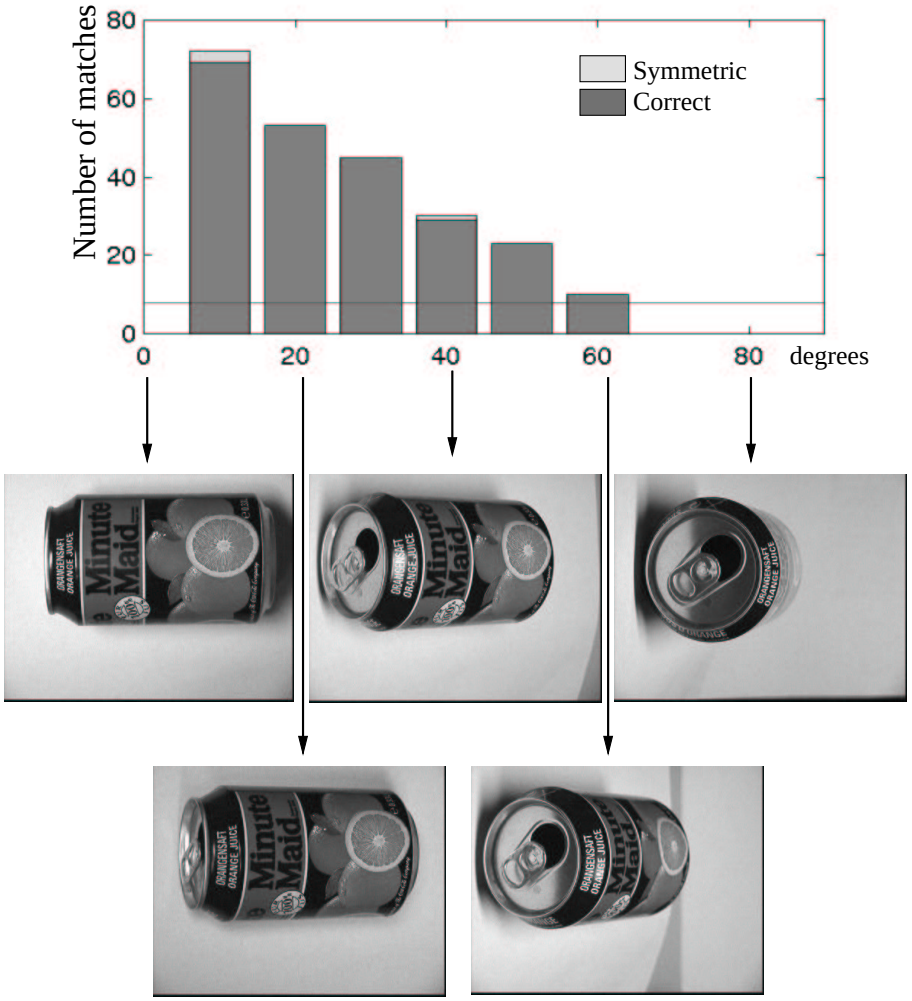


Figure 6.1: *Viewpoint invariance of the region extraction and matching: number of correct, symmetric and false matches found as a function of the rotation angle with respect to the reference view.*

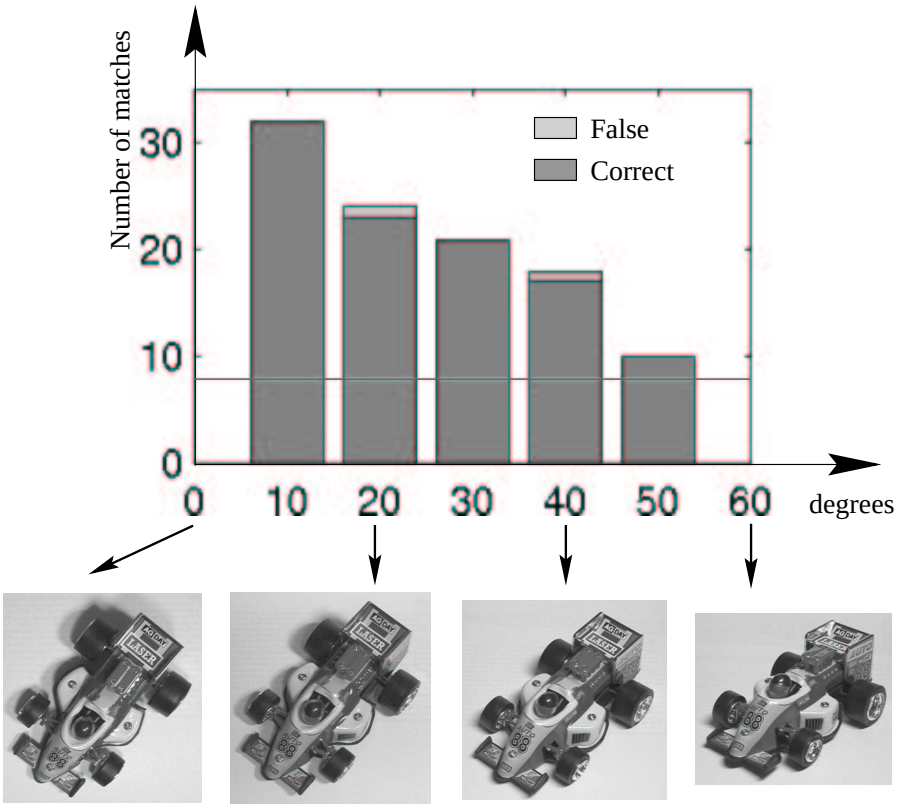


Figure 6.2: *Viewpoint invariance of the region extraction and matching: number of correct, symmetric and false matches found as a function of the rotation angle with respect to the reference view.*

The same experiment was repeated for another, more complex object, namely a toy car. The results of this experiment can be found in Figure 6.2. For this sequence of images, the system was not always able to reject the false matches. This is due to the fact that most good matches are more or less coplanar, such that the semi-local constraints checking for consistent matches are not very selective. Moreover, as the total number of regions was smaller, the number of matches was smaller as well. As a result, the system was robust only to rotations up to 50 degrees.

Robustness to scale changes

As scale changes seem to be the weakest point in the viewpoint invariance of the regions, we added an extra experiment to specifically test for the invariance to scale changes. For the same test object as used in the first experiment, images

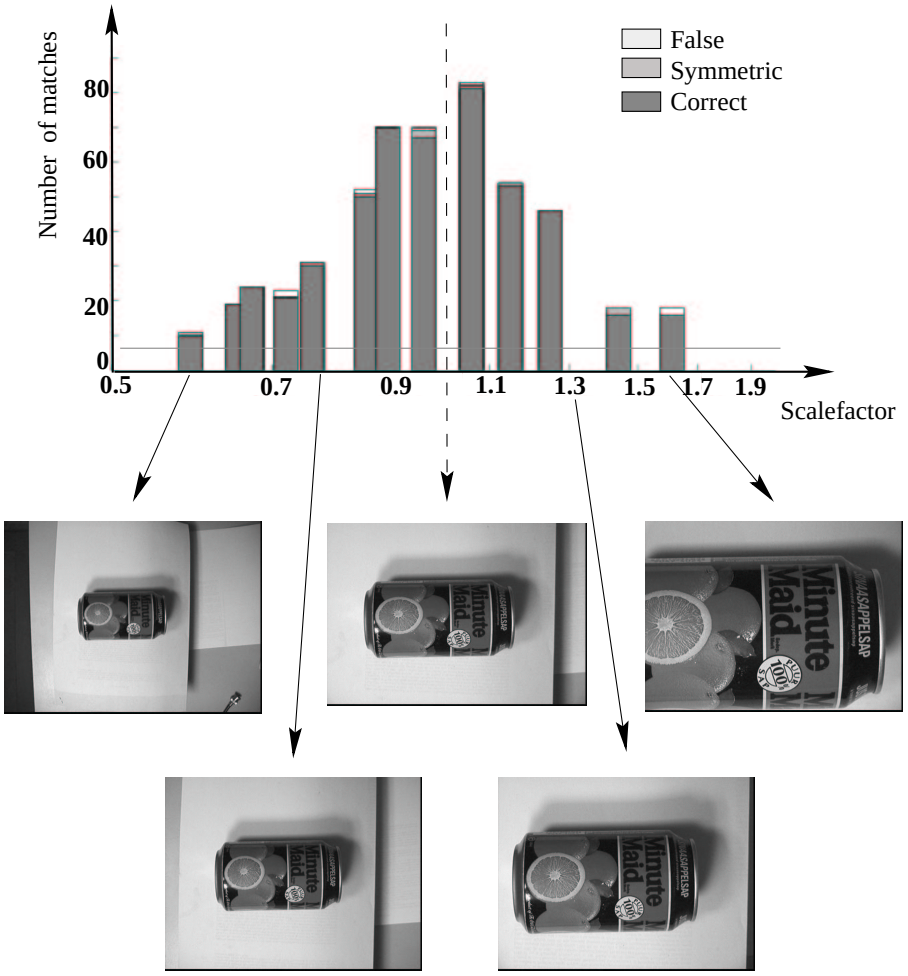


Figure 6.3: *Scale invariance of the region extraction and matching: number of correct, symmetric and false matches found as a function of the scalefactor with respect to the reference image.*

with different scales were taken by zooming in and out with our digital camera.

As can be seen from Figure 6.3, the number of matches found decreases with increasing scale change. Nevertheless, the extraction and matching of affinely invariant regions worked fine for scale factors ranging from 2/3 to 3/2.

6.2.2 Robustness to image blur

We also tested the robustness of the region extraction and matching to badly focussed cameras or image blur. To this end, each of the color bands of the

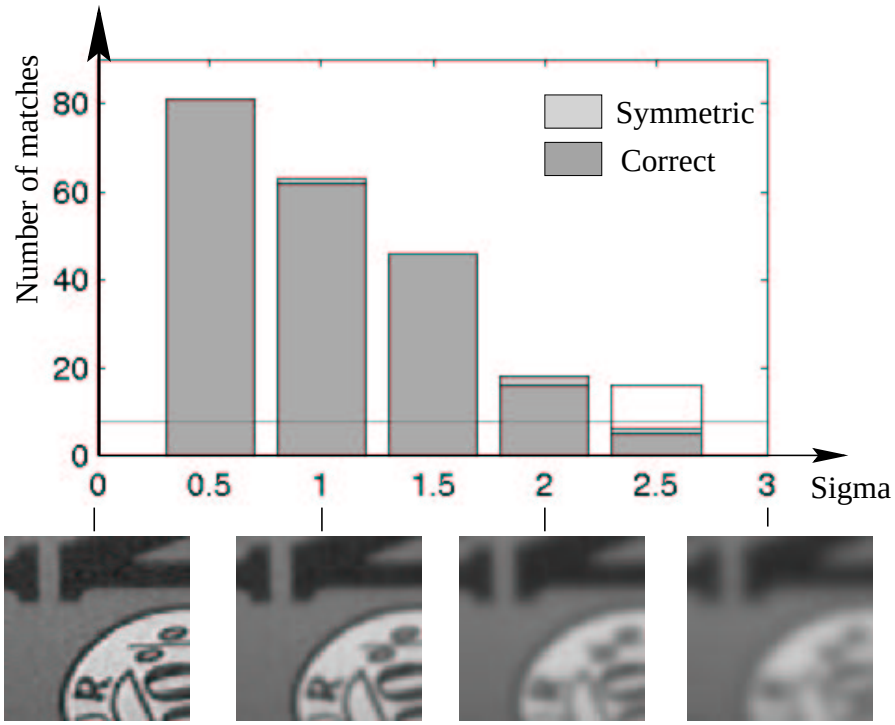


Figure 6.4: *Robustness of the region extraction and matching to image blur: number of correct, symmetric and false matches for increasing levels of Gaussian smoothing.*

reference image of Figure 6.1 was smoothed separately, using Gaussian kernels of increasing size.

The results of this experiment are summarized in Figure 6.4. Looking into more detail at the reason why this test performed so poorly, we found that it was the invariant feature vector which was most affected by the smoothing. Often, the same regions were detected in both images, but they were not matched because of the large Mahalanobis distance between the corresponding feature vectors. For the image smoothed with a Gaussian kernel with sigma equal to 2.5, for which only 5 correct matches and one symmetric match were found, we repeated the matching experiment, this time using a significantly higher threshold for the Mahalanobis distance. This resulted in much better results, namely 16 correct matches instead of 5 and no symmetric or false matches left (with an average Mahalanobis distance of 2.6). So we can conclude that the affinely invariant region extraction as such is rather robust to image blur, but the invariant description and hence the matching algorithm is not. Maybe we put our thresholds too strict. Another explanation could be that image blur was

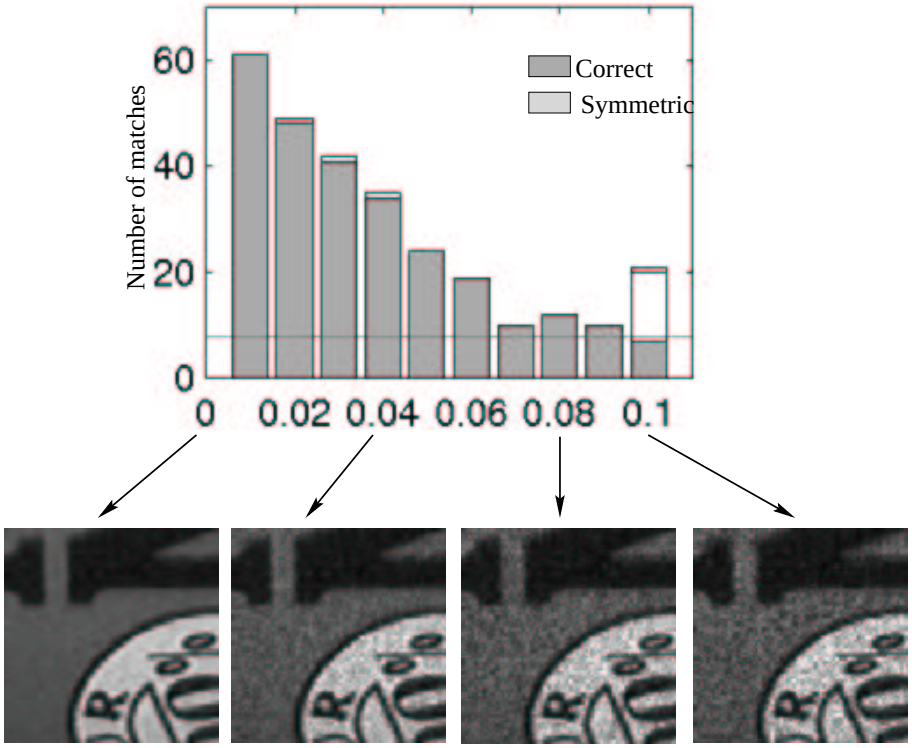


Figure 6.5: *Robustness of the region extraction and matching to image noise: number of correct, symmetric and false matches for increasing levels of multiplicative noise.*

not learned to the system when computing the statistics for the Mahalanobis distance.

6.2.3 Robustness to noise

To quantitatively check the robustness of our method to image noise, we added multiplicative noise to each of the three colorbands of the image, using the equation

$$I' = I + nI \tag{6.1}$$

where n is uniformly distributed random noise with zero mean and increasing levels of variance between 0.01 and 0.1.

The results of this experiment are summarized in Figure 6.5. In this experiment, we noticed the same effect as for the robustness to image blur, i.e. much better results were again obtained if we used a higher threshold value for the Mahalanobis distance than usual. For instance, for the image with noise

with a variance of 0.1, where we originally found only 7 correct matches, simply increasing the threshold value resulted in 20 correct matches and 1 symmetric one (with average Mahalanobis distance 2.5). Again, this can be explained by the fact that the artificial noise used in this experiment differs significantly from the variations included in the statistics for the Mahalanobis distance.

6.2.4 Robustness to illumination changes

Since changes in the illumination are harder to quantify than changes in scale, viewpoint, noise or image blur, we used some of the images provided by Funt *et al.* [FBM98] to test the invariance of our system to this kind of changes, as they provide very detailed information on the different illuminants used. Using these images, which are readily available through `ftp` [1], allows for easy comparison of our results with other recognition systems. Figure 6.6 shows the result. Each of the images shown below the horizontal axis was compared with the reference image taken under halogen illumination shown at the top of the figure. The left part of each image shows the white to black row of the Macbeth Colour Checker, highlighting the large difference in illumination. Most of the 'symmetric' matches found were actually matches between these reference squares. For all images, plenty of correspondences were found, clearly showing the robustness of our region extraction and matching to changing illumination conditions.

6.3 Conclusion

Experiments were conducted to show the robustness of the local invariant region extraction and matching to changing viewpoint, scale changes, noise, image blur and changes in the illumination. Generally speaking, the overall system works reasonably well under changing viewing conditions. As long as the variations in the images are not extreme, a fair amount of matches can be found, with only a minority of false matches, mostly caused by symmetries in the image.

From these experiments, we can conclude that our system can be used for object recognition purposes with only a limited number of reference views. For most objects, we estimate that ten or twenty images suffice to build a model under changing viewing conditions. If large changes in scale are to be expected, reference images of different scales should be included as well, as this seems to be the weakest point of the viewpoint invariance.

[1] For more information about these images, we refer to the authors' website: `http://www.cs.sfu.ca/~colour/image_db/index.html`

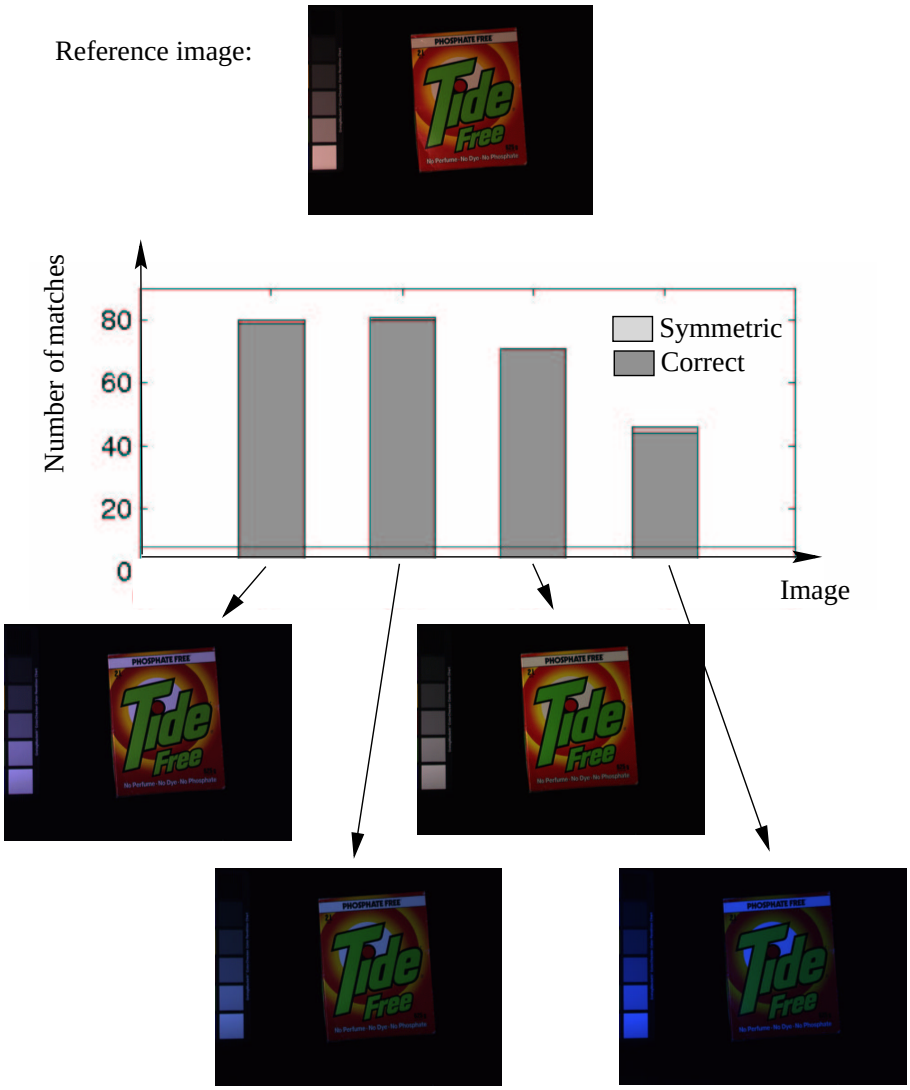


Figure 6.6: *Illumination invariance of the region extraction and matching: number of correct and symmetric matches found between the images shown along the horizontal axis and the reference image shown on top.*

Chapter 7

Wide baseline stereo

In this chapter, the application of matching affinely invariant regions between images to wide baseline stereo is discussed, i.e. the processing of two (or more) images taken from substantially different viewpoints. After a short introduction, referring to some other attempts to solve this problem (section 7.1), we continue with some experimental results in section 7.2. Section 7.3 concludes the chapter.

7.1 Introduction

Registration, or finding correspondences, between two widely separated views is still an important open issue in computer vision. Albeit a straightforward task for humans, computers seem to get lost once the change in viewpoint becomes larger. Not only can a corresponding point lie about anywhere in the other image – or even disappear –, also its appearance can change drastically due to geometric and photometric deformations.

Nevertheless, for many applications, the ability to work with images taken further apart from one another brings some clear advantages. Reconstruction of a scene in 3D is a good case in point. Working with images looking at the same scene from substantially different viewing angles significantly increases the accuracy of the result. Similarly, a substantial rotation around the optical axis between two images allows a better self-calibration of the camera, as the aspect ratio can be determined more accurately.

At the same time, also the flexibility of a system increases if it can deeal with wider baselines. During the acquisition of images, one no longer has to bother about the distance between two adjacent images. Sometimes, taking a sequence of small baseline images is even completely impossible for practical reasons. Also, older film material can be used (e.g. for 3D reconstructions), without the restriction to material consisting of images taken from more or less the same viewpoint.

In this section, we focus on the automatic extraction of the epipolar geometry in such wide baseline image pairs. To this end, a sufficient amount of correspondences between the images is needed, which we extract by matching local affinely invariant regions. Robustness to false matches is obtained by using the semi-local constraints proposed in section 5.5, and by applying RANSAC [FB81] for the derivation of the epipolar geometry. From this epipolar geometry it is possible to derive some information concerning the relative orientation and position of the two cameras (if the internal camera parameters are known). This will be exploited in section 10.2 in the context of visual homing of a mobile robot. Still, the most important advantage of knowing the epipolar geometry is that it makes it easier to find additional correspondences, as the search for a corresponding point is now limited to the epipolar line. Alternatively, the images can be rectified using the epipolar geometry, such that corresponding points lie on the same horizontal line. In this way, the effect of viewpoint changes can partially be compensated for.

Related work

A straightforward approach to wide baseline stereo matching would be to deform a patch of the first image in an iterative way, until it more or less fits a patch in the second image (e.g. [SK97]). In contrast, our method is based on the extraction and matching of the local invariant regions and hence works on each of the two images separately, without searching over the entire image or applying combinatorics.

This approach is akin to that of Pritchett and Zisserman [PZ98a, PZ98b] who start their wide baseline stereo algorithm by extracting quadrangles present in the image and match these based on a normalized crosscorrelation to find local homographies, which are then extended to larger parts of the image. However, they use shapes that are explicitly present in the image, while ours are determined locally based on the color or intensity patterns around anchor points and do not rely on specific structures in the scene. Hence, the applicability of our method is much wider.

Recently, Tell and Carlsson [TC00] also proposed a wide baseline correspondence method based on affine invariance. They extract an affinely invariant Fourier description of the intensity profile along a line connecting two corner points. The non-local character of their method makes it more robust, but at the same time restricts its use to planar objects, which is a serious limitation on the applicability of their method.

Finally, Georgis *et al.* [GPK98] use a Randomized Hough transform-like version of the eight-point algorithm. However, for their method to work, they need four coplanar points which are manually selected, so their method is not fully automatic.

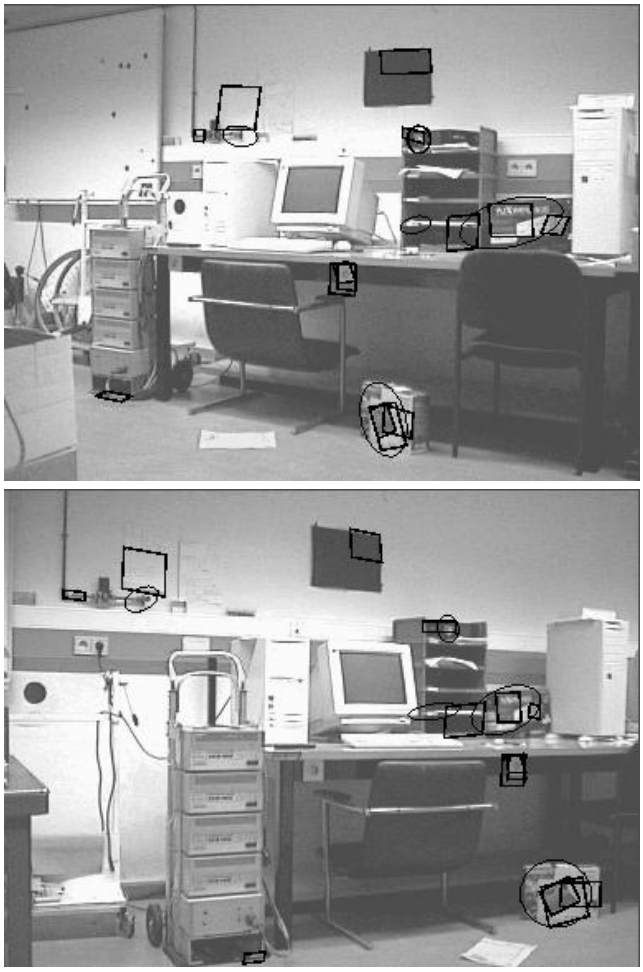


Figure 7.1: *Example 1: Final region correspondences (all matched correctly).*

7.2 Experimental results

Figures 7.1 and 7.2 show a first example. Although the distance between the two cameras is about 4 or 5 meters, and the change in orientation is over 50 degrees, several correct region correspondences could be found (after the filtering, using the geometric and photometric constraints and testing the epipolar geometry using RANSAC). These are shown superimposed on the images in Figure 7.1. All these regions are matched correctly to the corresponding region in the other image. However, as can be seen in the figures, not all region correspondences match perfectly. This is due to the lack of texture within these regions (e.g. the quadrangle on the wall), or due to the non-planarity of the



Figure 7.2: *Example 1: Epipolar geometry.*

neighborhood of the interest point (e.g. some regions found on the rack). Nevertheless, the similarity between the regions was good enough for these matches to be found and to be maintained throughout the filtering process.

From those region correspondences, the epipolar geometry was derived using RANSAC. Some corresponding epipolar lines are shown in Figure 7.2. These clearly correspond very well over the entire image.

As a second wide baseline example, consider the images shown in Figure 7.3. The camera has not only been rotated between these two images; there is also a substantial translation, as for the left image the camera was facing straight ahead while for the right image we are looking down onto the scene. The affinely



Figure 7.3: *Example 2: Final region correspondences (all matched correctly).*

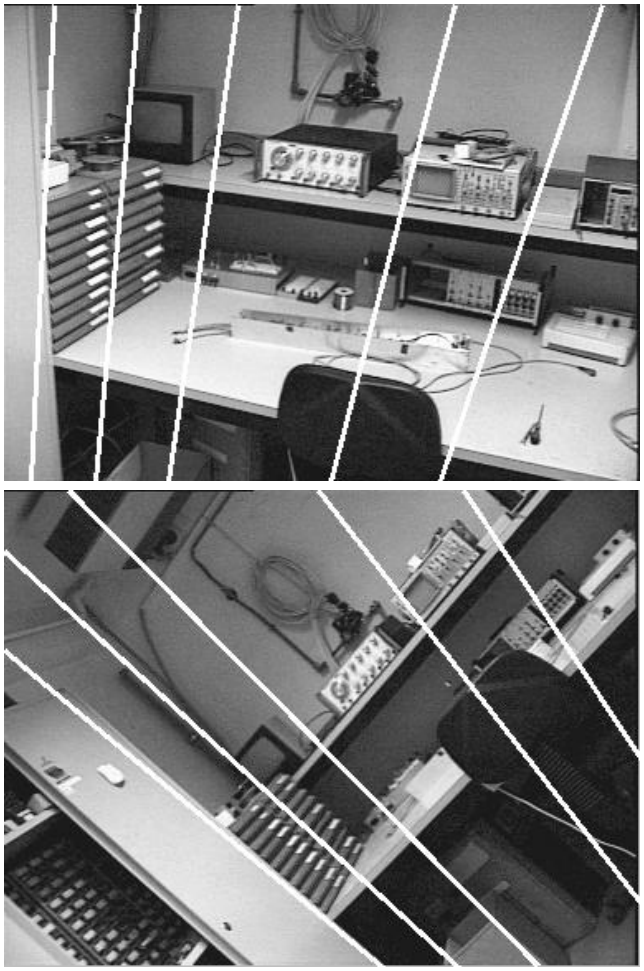


Figure 7.4: *Example 2: Epipolar geometry.*

invariant regions matched between those two views are shown superimposed on the images, while the resulting epipolar geometry can be found in Figure 7.4.

As a third and last wide baseline stereo example, look at the two images shown in Figure 7.5. Note the large change in viewpoint, resulting in large scale changes in some parts of the images, and extreme foreshortening in other parts. Nevertheless, a sufficient number of matches was found for an accurate determination of the epipolar geometry (shown in Figure 7.6). At the same time, the two false matches in Figure 7.5 were correctly rejected by RANSAC, as they were not consistent with the epipolar geometry.



Figure 7.5: *Example 3: Final region correspondences (black for correct matches, white for false matches).*

7.3 Conclusion

From the examples given in this chapter, it becomes clear that the proposed affinely invariant regions can be used to extract the epipolar geometry even for images taken from substantially different viewpoint, as long as the image content is sufficiently 'rich' (i.e. it should contain enough textured surfaces to find regions on) and as long as there is not too much repetition or symmetry in the images, causing too many false matches. For most images, this is not a problem though.

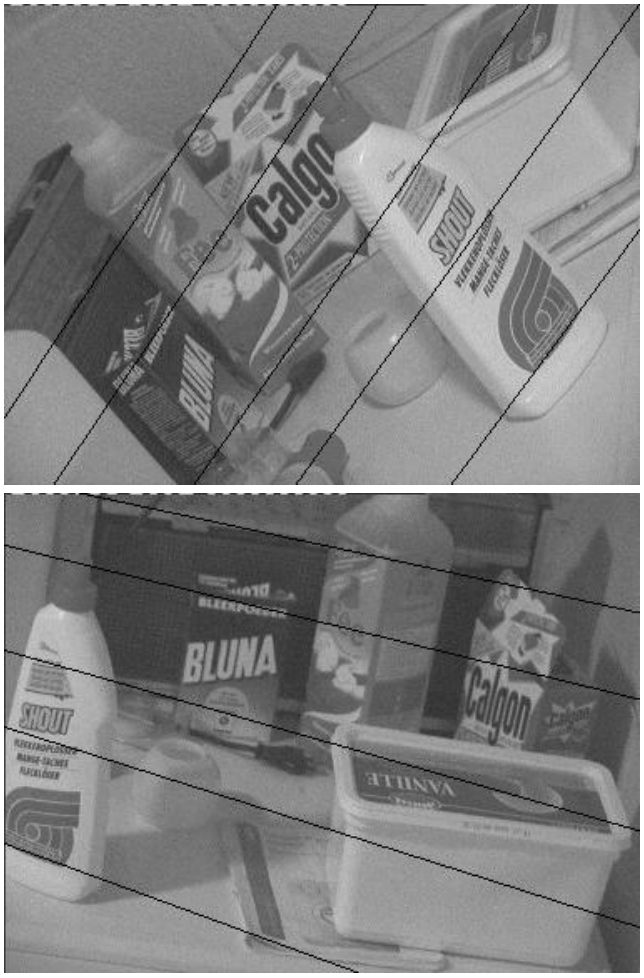


Figure 7.6: *Example 3: Epipolar geometry.*

Chapter 8

Object recognition and localization

Although the primary motivation and inspiration for this work stems from the intriguing domain of object recognition, the focus up to this point has been on the correspondence problem, i.e. finding corresponding regions between two images of the same object. In this chapter, we explain how the algorithms for local invariant region extraction and matching can be used to describe, recognize and localize objects. We start with a short introduction (section 8.1). Next, in section 8.2, we explain how the affinely invariant region extraction and matching can be used for object recognition. In section 8.3, some experimental results are given. Finally, section 8.4 concludes the chapter.

8.1 Introduction

The object recognition problem is essentially this: given some knowledge of how certain objects may appear, plus an image of a scene possibly containing those objects, find out which objects are present in the scene and where.

Recognizing and localizing objects is an important capability for many automated systems. It is important for a robot picking parts from a conveyor belt, for automatic inspection, and for an intelligent system trying to understand its environment (e.g. robotic hazardous waste cleanup). Although commercial solutions exist for simple recognition problems, they depend on carefully controling the environment to eliminate the confounding effects of changing background, image clutter, occlusions and lighting and viewpoint variations. With the local, affinely invariant regions proposed in this work, a wide range of objects as encountered in every day life can be recognized under much more general viewing conditions.

We have to admit though that, to date, we have not run any large object recognition experiments yet. Instead, most of our experiments have been

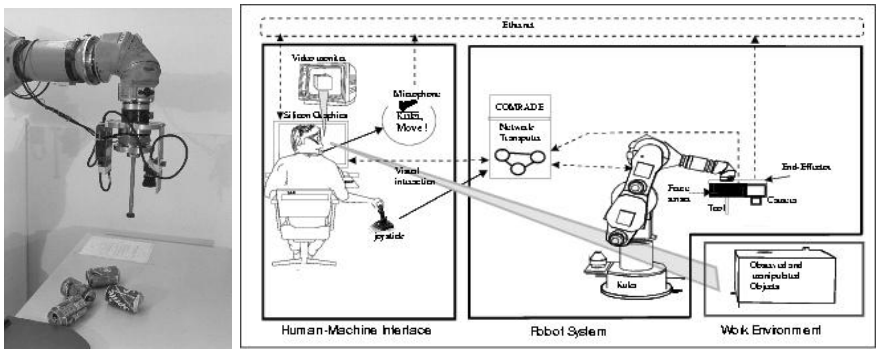


Figure 8.1: *Object recognition and localization as part of an integrated supervisory control system*

limited to five objects or even less, which makes it more an *object localization* task than real object recognition, i.e. try to find out whether a specific object is present in the image or not, and if so, where in the image it is located. Nevertheless, we believe that the feasibility of larger object recognition experiments is sufficiently demonstrated through the wide baseline stereo and database retrieval experiments we performed (see chapter 7 and chapter 9).

Part of these object localization experiments have been performed in collaboration with the mechanical department of our university [ZaaAl00]. In this context, the object recognition and localization are one module of a larger, *Integrated Supervisory Control System* or ISCS for short. This is a human-machine interaction scheme using multi-modal interaction (both vision, speech and telemanipulation) and multi-level control (ranging from simple telemanipulation over shared control levels to (practically) full autonomy) [ZVB99]. Applications of such systems include tele-operation in a nuclear power plant or in a space mission as well as giving more autonomy to the physically disabled. For an overview of the system, see Figure 8.1.

In these experiments, the recognition and localization of the object in the image is followed by a localization of the object in 3D based on stereovision, and some form of interaction with the object (touching, grasping, ...) by the manipulator.

8.2 Object recognition using local invariant regions

For each object we want to recognize, several model images taken from different viewpoints are collected. Next, the local affinely invariant regions are extracted from each of these images and described by a feature vector of moment invariants. All the feature vectors are then stored in a database, together

with a pointer to the image from which they originate and the coordinates of the corresponding invariant region. Up to this point, the computations can be done off-line, i.e. before a specific query has been made, as all processes up to this point are independent of the query image.

When a query image is then passed to the system, one or more model image(s) of the object(s) present in the query image are retrieved from the database using a *voting mechanism*. This goes as follows. We extract the invariant regions from the query image and compute the corresponding feature vectors of moment invariants. For each feature vector belonging to a region in the query image, similar feature vectors describing similar regions extracted from any of the model images in the database are selected. This can be done in an efficient way using indexing techniques, making the object recognition less dependent on the number of modeled objects in the database. The Mahalanobis distance between the feature vectors is first evaluated. If the Mahalanobis distance is below a predefined value, then the crosscorrelation between the two (normalized) regions is checked. If the crosscorrelation is above a predefined value, the match is 'accepted', and a vote is attributed to the corresponding model image. In this way, all accepted matches take part in a voting process to finally rank the model images of the different objects according to similarity (i.e. in order of decreasing number of corresponding regions with the query image).

If images of an object, present in the query image, are stored in the database, regions on the object will be matched to the corresponding regions found in those model images, while votes from regions on unknown objects or from the background will be randomly spread over all the database images (if they attribute votes at all). As a result, the images of the corresponding object will probably get more votes than the other model images.

As the number of extracted regions may vary by a factor 10 depending on the image content, simply selecting the image that has received the highest number of votes is not optimal. Indeed, images with many regions are more likely to get a high number of votes due to accidental similarities. To compensate for this effect, we look at the relative number of votes, i.e. the total number of votes for a specific model image divided by the total number of regions found in that image.

All images with a significant number of votes are selected as matching candidates. The matches for these images are then further tested using semi-local constraints (see section 5.5). If they survive the consistency-tests, we can be pretty sure that the corresponding object was indeed present in the query image.

8.3 Experimental results

As a test case, we ran a number of experiments, where the goal was to recognize and localize different drink cans in a cluttered environment. Five different types



Figure 8.2: *Reference views modeling a coke can*



Figure 8.3: *Based on the model images of Figure 8.2, the coke can is correctly localized.*

of cans (with similar shapes but different textures) had to be distinguished. One of them was a coke can. Hence, several images of the coke can from different views were collected, to model the object. All the model images for this object are shown in Figure 8.2. These seven images are sufficient to model the coke can from a wide range of views. Images from the top and bottom of the can were omitted, as these were not distinctive in this application, where all modeled objects were different makes of drink cans. Then, given a new image, the system tries to match the affinely invariant regions for this image with the affinely invariant regions for all the model images. If there is no image that gets a sufficiently high number of votes, the system decides that the image does not contain any relevant object. Otherwise, the object has been recognized. Localization is performed by looking at the position of the matched affinely invariant regions. An example of correct object localization of the coke can is shown in Figure 8.3. Note that the parallelogram added to this figure gives just a rudimentary localization. This parallelogram is the backprojection of the



Figure 8.4: *Corresponding regions used to localize the coke-can*

boundaries of the most similar model image (the most left one in Figure 8.2), computed based on the average affine transformation of all consistent matches for that image. This, in fact, only gives a valid result if applied to planar objects. Figure 8.4 shows the local affinely invariant regions that have been matched between the image of Figure 8.3 and the most similar model image.

The system was able to detect and distinguish several different drink cans, even in cluttered scenes, from a wide range of viewpoints. Only in case of strong specular reflections on the cans the detection sometimes failed, wrongfully concluding there was no relevant object in the query image.

8.4 Conclusion

Using a voting mechanism, the affinely invariant region extraction and matching can be used in the context of object recognition and localization. Each object is modeled by the affinely invariant regions extracted from a set of model images taken from different viewpoints. Then, each invariant region found in the query image attributes a vote to all model images containing similar regions. In this way, model images belonging to objects present in the query image will typically get a higher number of votes than the other model images, such that the object(s) can be recognized. At the same time, the object has been localized within the image, based on the position of the matching regions.

Chapter 9

Content-based image retrieval

This chapter on content-based image retrieval is organized as follows. First, a brief introduction and a short overview of the state of the art in content-based image retrieval is given in section 9.1. Next, we have a closer look at how we can use the affinely invariant region matching to retrieve similar images from a database, using a modified interpretation of 'similarity', slightly different than usual in the database retrieval literature (section 9.2). Section 9.3 gives the experimental results on a database of 465 images, while section 9.4 discusses a related application, namely a virtual museum guide. Section 9.5 concludes the chapter.

9.1 Introduction

Due to the explosive growth of easily accessible digital information over the internet over the last ten years, the problem of data-mining is steadily gaining importance. One important and new aspect to this problem is the fact that much of the digital information exists under the form of image- and video-material. As a result, *content-based image retrieval* has become a hot research topic within the computer vision community. It addresses the problem of designing tools for searching, browsing and automatically annotating images and image-sequences based on visual information (as opposed to text-based approaches that use manually added text-annotations).

In a survey, Picard [Pic95] identified three types of requests in stock photo searches: requests concerning subject and action content (e.g. images of three kids playing together), important for editorial purposes; requests based on picture syntax (e.g. images with horizontal structures, open smooth surfaces) and requests concerning subjective components (mood - e.g. frolicsome, bright, autumnlike), important in commercial photography for advertising. The really

challenging questions are the ones that query for semantic-rich descriptions. As a result, content-based image retrieval seems more related to *image understanding* than to low level feature extraction.

Nevertheless most practical implementations of content-based image retrieval to date focus on the general appearance of an image, using statistical information about its color or texture rather than concentrating on the semantics. Such global image features, although quite successful for a surprisingly large number of queries, are nevertheless restricted in their usefulness. They are less effective when e.g. looking for other images that contain the same object or simply the same scene but seen from a very different viewpoint. Relative amounts of color and texture typically change dramatically in such cases. Hence, more local features are needed as well, that capture local image structure in a way that is robust against changes in viewpoint and illumination. The local, affinely invariant regions developed in this work seem well suited for this task.

Related work

Most existing image retrieval systems mainly focus on color and color distribution. A very popular and well-developed technique is based on color histograms, first developed by Swain and Ballard [SB91], refined by Funt and Finlayson [FF95], as well as by Healey and Slater [HS94] to obtain illumination invariance (see chapter 2). Others have tried to use texture (e.g. [LP95]) or shape (e.g. [MAK96, DP96]), but the use of these criteria is mostly limited to a specific application. Some commercial systems are available as well, combining both color, texture and shape, depending on the user's query [QBIC95, Virage97]. However, as already mentioned in the introduction, all of these are mainly based on *global* image features.

Maybe the most similar work in the context of content-based image retrieval is that of Siggelkow and Burkhardt [SB98]. They extract local invariant features for each pixel in the image and store these in a histogram, which can then be compared with a similar histogram for each image in the database using histogram intersection techniques. The use of histograms makes the system robust to partial views and object occlusions. The local invariant features on the other hand add some structural information to the image description and allow to cover a larger class of transformed images with only one histogram. However, invariance in their case is limited to 2D translation and rotation only. Another difference is that they do not search for individual correspondences but rather consider the overall statistics only, using histograms.

9.2 Content-based image retrieval based on local, affinely invariant regions

From now on, we define two images to be similar if and only if they contain *identical* objects. This, in turn, can be checked by counting the number of corresponding affinely invariant regions found in the images.

Note that we said *identical* objects and not just *similar* objects. Indeed, we look at *local* regions of interest on the objects, so we compare details of the objects rather than the overall appearance. As a result, a global similarity of objects is not sufficient. They should be identical up to the smallest level of detail. In this respect, it might be better to use the notion of *relevance* instead of *similarity*. Indeed, an image containing the same objects as the ones shown in the query image is maybe not the most similar on a global scale (e.g. due to changes in illumination or viewpoint, image clutter or changing background). However, in some situations (depending on the query), it is the most relevant. For instance, if one poses a query by showing an image of ones favorite actor or the president of the United States, one is not interested in images of other people in more or less the same pose, but rather in other images of the same individual. The same holds for query images containing specific buildings, monuments, places etc.

The actual database retrieval is very similar to the object recognition system described in the previous chapter. Again, all local invariant regions found in the query image take part in a voting mechanism, attributing votes to the images in the database from which similar, local invariant regions have been extracted. The images that get the highest number of votes have the highest probability of actually containing the same objects as present in the query image. Finally, all images with a significant number of votes (relative to the maximum number of votes attributed to an image) are further tested using the semi-local constraints, to obtain a final ranking of the database images.

9.3 Experimental results

We have tested our algorithm on a database of 465 color images, mainly consisting of manmade objects (both indoor and outdoor). Most of these images are of size 576×768 pixels (i.e. PAL resolution). For 56 of these images, at least one extra image taken from a different viewpoint and/or under different illumination conditions was available to be used as query image. Most of these images were recorded in our lab or in the city of Leuven, with a Sony digital camera. To these, images from the Corel Professional Photos database selected from the packages "Big Apple", "Egypt", "India" and "China" and images from the California Departement of Water Resources database [DWR] were added to end up with a total of over 450 images of different objects or scenes. The images from the `Corel` and `DWR` databases were selected in such a way that the resulting database stayed as homogeneous as possible.



Figure 9.1: *Some examples of query images (top of pair) and their corresponding images in the database (bottom of pair) where the image retrieval was successful.*

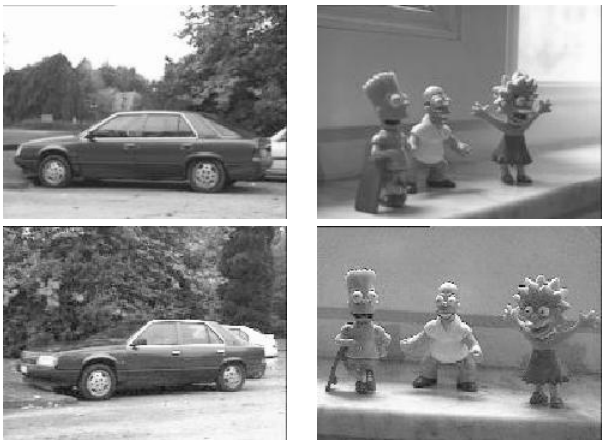


Figure 9.2: *Some examples of query images (top of pair) and their corresponding images in the database (bottom of pair) where the image retrieval failed.*

Figures 9.1 and 9.2 show some query images (upper rows) together with the image of the same object or scene in the database (lower rows). For all the examples of Figure 9.1, the corresponding database image was correctly retrieved from the database, i.e. the correct image got the highest number of votes. Note that there often is a substantial change in viewpoint between the query image and the corresponding database image. Also the background may have changed drastically, such as in the lower middle example. This has hardly any influence on our system, but may cause problems for systems that look for global similarities.

Figure 9.2 on the other hand shows some examples of images where our system failed to retrieve the corresponding image from the database. For the car images, this is mainly due to the lack of texture on the car, making it hard to extract sufficiently discriminant regions, combined with specular reflections on the windows and different backgrounds. For the other image pair, the main reason for its failure is again the lack of texture, combined with the fact that most edges in this image are actually unstable object contours. Too many occlusions, such as might happen when looking at a building behind a fence, can also cause serious problems during image retrieval.

Nevertheless, the correct image was retrieved in two out of three query images as the image with the highest number of votes. In 90 % of the cases, the correct image was among the top fifteen of retrieved images. Figure 9.3 gives an overview of the retrieval results. It shows the percentage of query images for which the correct image was the first, among the first five, among the first ten, and so on.

For a peculiar subset of the query images, the system was able to retrieve the correct database image although only a minority of the individual matches

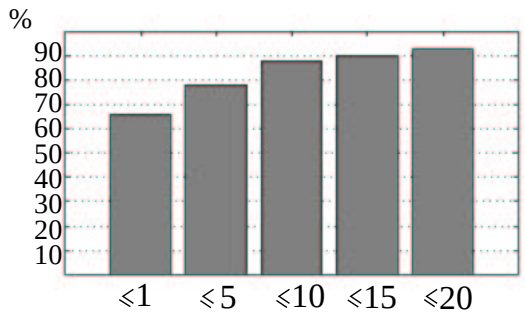


Figure 9.3: *Query results. In y percent of the queries, the correct image was ranked among the first x of retrieved images.*

were correct. Many of the corresponding regions in the query image and in the retrieved database image did not represent the same physical point, but were matched to other regions in the correct image, with more or less similar neighborhoods. This happens for instance when there is a repetition of the same pattern in the query image, such as the different windows of the same building or the different tiles of the same floor, which are all alike. The same phenomenon also occurs in the case of textured images. For instance, a tree-like texture on a large part of the image will rarely give exact matches (as trees are usually far from being planar objects). Nevertheless, it often results in a high number of votes for the correct database image, as this image has the highest probability of containing regions with a similar texture. This is in particular true for the elliptic, intensity-based regions. Of course, these cases do not always survive the semi-local constraints test.

This is reminiscent of the work of Schiele *et al.* [SC96], who perform object recognition based on local features not by searching for correspondences, but by computing the probability density functions of the features for specific views of each object and deriving the probability that the query image contains the object.

As a last remark, we would like to note that the results obtained on this database of 450 images are very similar to those obtained when our database was only 50 or 100 images large. This indicates that our results are rather insensitive to the size of the database. Further experiments on larger databases are necessary to confirm these results though.

9.4 An application: a virtual tour guide

As a demo-application of the database retrieval system, a palmtop assistant has been developed that gives information about paintings at the Van Dijck exhibition organized in the Antwerp museum of fine arts in remembrance of the painter's 400th birthday. The system consists of a hand-held PC with built-in

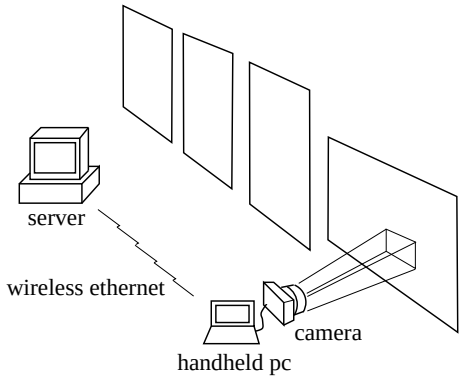


Figure 9.4: *Overview of the system's architecture.*

camera (e.g. Sharp's HC 4600 A) or a separate digital camera connected to the hand-held PC via an infrared link. Images are taken of paintings or details thereof and sent to a server via a wireless ethernet connection, where they are processed such that the desired information can be returned to the user. Figure 9.4 gives an overview of the system.

The server recognizes the painting the visitor is pointing the camera at. The visitor can also select details for more specific information. In fact, the server performs database retrieval to recognize the painting independent of the viewpoint, image clutter, occlusions (e.g. other visitors walking in front of the camera), changing illumination, etc. Once the content of the picture has been recognized, the appropriate information is returned to the user over the same wireless ethernet connection.

An advantage of such a system - apart from its intuitive user interface - is that it can derive from the picture what the user is actually interested in and adapt the returned information accordingly. This is difficult to achieve with e.g. electronic tour guides that use the viewer's location as primary input (e.g. using infrared transmitters placed in the ceiling above each piece to be described [Bed95]). The visitor is also free to roam the museum and ask for information in whatever order, which is difficult with tape recorders. Also, since all that is needed is a picture of the piece, no external adaptations have to be made. This makes the basic approach equally suited for less controlled environments like a moving exhibition, an open air show, or even a city tour.

Probably the most similar system is DyPERS (or Dynamic Personal Enhanced Reality System) developed by Schiele *et al.* [SchAl99], which has (among other applications) also been evaluated in a museum-gallery scenario. However, they recognize objects based on multidimensional receptive field histograms, which is probably less robust to changes in viewpoint than our system.



Figure 9.5: *An overview of our database for the virtual museum guide.*

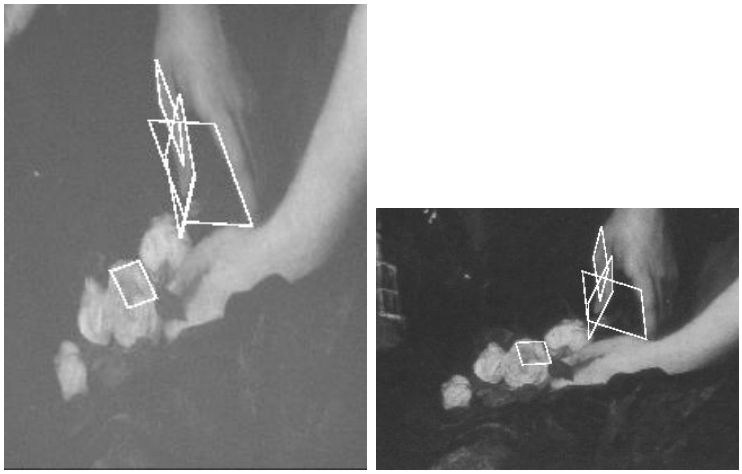


Figure 9.6: *A detail of the database image "The Ages of Man"(right) and of the query image (left) with corresponding affinely invariant neighborhoods.*

Experimental results Due to organizational problems as well as security considerations, we were in the end not able to evaluate our system at the real Van Dijck exhibition. Instead, we created a small mock-up gallery in our lab based on 50×70 cm^2 reproductions. It is important to note that this has complicated rather than simplified matters. The database images were scanned from a catalogue with high quality reproductions. Most posters in the mock-up gallery were of much lower quality, however, leading to a loss of color.

Images of these posters were matched to high-resolution (1000×1500 pixels) digital images of 20 of Van Dijck's paintings stored in our database. An overview of the database is given in Figure 9.5. In total, 12.978 affinely invariant regions were found in these 20 images, i.e. an average of 650 regions per image. These are all geometry-based regions, as the extraction of the intensity-based regions would make the system much too slow for this application.

The image on the right of Figure 9.6 is part of the database image "Ages of Man" (also shown in Figure 9.7). The image on the left is part of a query image of the corresponding poster, taken from a different viewpoint. Although the illumination conditions have changed drastically and the image is distorted due to the change in viewpoint, the affinely invariant regions found in both images correspond, i.e. they are deformed in such a way that they still cover the same physical content in spite of the change in viewpoint.

For a first example of image retrieval, look at the painting called "The Ages of Man", shown in Figure 9.7. At the top, the database image of the painting is shown with some possible user queries just below. Note the large differences in color and the affine image deformations. Not visible in this rescaled figure is the difference in scale, which varied in between 80 and 130 percent. Nevertheless,



Figure 9.7: *Given the query images shown below, the system was each time able to retrieve the correct database image (i.e. to identify the painting out of the list of 20) and to highlight the corresponding part therein (top).*

the correct painting was retrieved from our database for each of the three query images. On top of the database image, parallelograms are added. These represent the corresponding image part as computed by the server, through backprojection of the query image boundaries.

Similarly, a second example based on the painting "Maria Louisa de Tassis" is shown in Figure 9.8. Many more experiments were conducted on this dataset. In general, the correct painting was always correctly retrieved, unless if the user zoomed in on a part of the painting not containing enough information, such as a dark background region, or if the change in scale was too large. However, the latter problem can easily be solved by adding model images at different scales to our database, while the former is solved in a natural way, as visitors are usually only interested in those parts of a painting containing a sufficient amount of information.

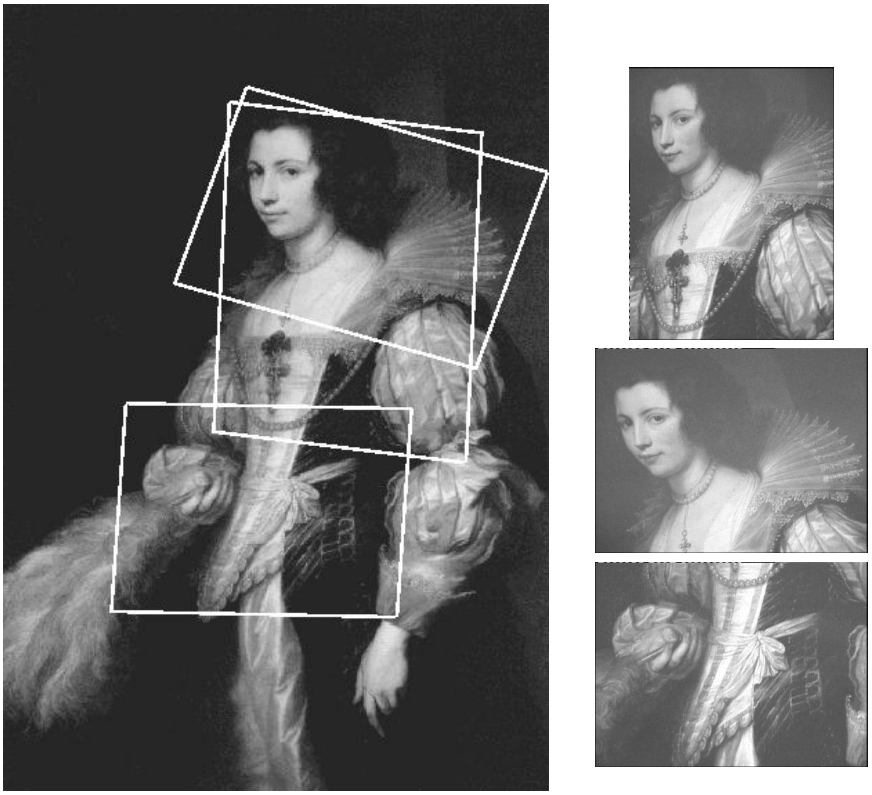


Figure 9.8: *Given the query images shown on the right, the system was each time able to retrieve the correct database image (i.e. to identify the painting out of the list of 20) and to highlight the corresponding part therein (left).*

9.5 Conclusion

In this chapter, the feasibility of using *local* features for content-based image retrieval has been demonstrated. Here, we look at image retrieval from a different perspective than usual. Instead of looking for some global similarity between images such as color or texture statistics, we rather focus on the objects present in the query image and try to retrieve images from the database representing the same objects. Using this restricted definition of similarity, promising results have been obtained.

Chapter 10

Mobile robot navigation

In this chapter, the use of local, affinely invariant regions in the context of mobile robot navigation is discussed. First, a short introduction to vision-based mobile robot navigation is given (section 10.1). Next, section 10.2 describes how our matching algorithm can be used by a mobile robot for visual homing. Given a target image, the robot can automatically go to the corresponding position by matching affinely invariant regions extracted from the image it currently sees with the target image. Section 10.3 describes two possible strategies for a vision-based navigation algorithm. The first one mainly builds on the visual homing algorithm, while the second one uses the affinely invariant regions as natural beacons to guide the robot. Section 10.4 concludes the chapter.

10.1 Introduction

As the cost of digital cameras goes down year after year, while the processing power of PC's keeps going up, more and more mobile robots are equiped with one or more cameras as extra sensors to extract richer information from the environment than possible with the more traditional ultrasonic sensors or laser range finders found on most mobile platforms. However, capturing images of the environment is one thing. Extracting useful information from them is another. This task is even more challenging as most mobile robot applications work under unconstrained environments and – in contrast to most computer vision applications – require real time operation.

Nevertheless, some very successful examples of integration of vision and robotics have been realized. The most prominent example is probably *road following*. Using simple computer vision algorithms, experiments have been demonstrated where a robot or automated car is able to follow a road not only in an artificial environment but also on real highways, at speeds of up to 100 km/h or more (e.g. [BM98, Tho90]).

Another typical example of integration of vision and robotics is *visual servoing*. This is a rapidly maturing approach to the control of robot manipulators, based on the visual perception of robot and workpiece location. The camera is considered as a sensor measuring deviations in the image from desired marker positions. As such, the camera can directly be included in a control loop to position the robot's end-effector relative to the workpiece as required by the task. However, these systems still show some serious limitations as well: they mostly work with specific markerpoints, which are easy to detect, match and track and rely (implicitly or explicitly) on a 3D model of the workpiece. For an overview on this topic, we refer to [Cor96, Has93].

Intuitively, one feels that the use of cameras on mobile platforms can go much further. Similar to humans, we want our mobile robots to use vision for such complex tasks as finding a specific object, opening doors, localizing itself, recognizing persons, navigating through a building, and so on.

We believe that vision is not used to its fullest extent in this context. Too often, its use is restricted to low level feature extraction, such as the detection of edges or lines (e.g. [KK93]) or even artificial landmarks (e.g. [AH93]), and hence relies on strong assumptions about the environment. Alternatively, a stereo head is mounted on top of the robot (e.g. [KriAl93]), such that depth information can directly be extracted from the images, reducing the camera to a depth-measuring device that can be integrated in the existing navigation algorithms much in the same way as a laser range finder or ultrasonic sensors.

The affinely invariant region extraction developed in this work can be useful as an alternative to the model-based feature extraction methods and the (often too complex) 3D reconstructionist approaches to mobile robot navigation and place recognition. Due to their insensitivity to viewpoint and illumination changes, the affinely invariant regions can be used as *natural landmarks* or *beacons*, as opposed to artificial landmarks. They can be detected from a wide range of viewpoints, and are individually discriminant based on the feature vector of moment invariants. Due to their local character, they are robust to partial occlusions and scene changes as well.

The object recognition and localization system, described in chapter 8, can directly be used by any mobile robot to localize specific objects in its environment or to localize itself (e.g. after a power shutdown) by recognizing its environment. Moreover, more complex algorithms for visual homing and vision-based navigation can be built on the basis of the local affinely invariant region matching as well. How this can be done is the main topic of this chapter.

10.2 Visual homing

Visual homing is the problem of automatically having the robot return to a known position –its *home*– purely based on visual information.

In contrast to most other work on visual homing (e.g. [BR95, DZ96]) and similar work in the context of image-based visual servoing [Cor96, Has93], the

method we propose here is based *only* on the target image (i.e. an image taken at the target position). No 3D model of the environment or images of intermediate locations are required. The only assumption we make is that there is a sufficiently large overlap between the start image and the target image.

The target position is completely specified by the target image (assuming that the internal camera parameters are fixed and known). The basic strategy is then to match the image currently seen by the robot to this target image, and to navigate the robot to compensate for the viewpoint differences.

However, image features are in general not viewpoint invariant which makes it difficult to match the current image with the target image, unless if specific markerpoints are used. It is clear, however, that in many applications such artificial landmarks are undesirable, if not completely impossible. That's where our affinely invariant regions come in and offer a way out, resulting in an algorithm that does not need any special landmarks nor a priori (3D) knowledge about the scene and that can deal with large viewpoint changes. It is based on the matching of affinely invariant regions between views and an interpretation of the disparities and epipolar geometry in terms of relative robot position and orientation.

This is akin to the work of Basri *et al.* [BasAl98, BasAl99], on which our algorithm is partially based. However, they do not consider the problem of finding the initial correspondences. Tell and Carlsson [TC99], on the other hand, use line intersections and invariants derived from the geometric structure of the lines endpoints to find correspondences, which is probably less general than our method. Moreover, their visual homing algorithm can only deal with pure translations between views.

System overview In our system, the visual homing task has been divided in the following steps:

- Find correspondences between the start image and the target image.
- Compute the essential matrix $\mathbf{E}$.
- Determine the relative rotation $\mathbf{R}$ from $\mathbf{E}$.
- Determine the direction of the relative translation $\mathtt{t}$ from $\mathbf{E}$.
- Take a third image and determine the magnitude of the relative translation $\|\mathtt{t}\|$.
- Move to the desired position.

If the motion is too large, an iterative procedure may be applied with regular recomputations of the relative rotation and translation to prevent errors due to odometric inaccuracies. While the robot is moving, the points that are matched to the points in the target image may be tracked continuously (as in our implementation) or, alternatively, new correspondences may be searched for whenever needed.

Each of the steps mentioned above will now be discussed in more detail, except for the last one, which completely depends on the specific robot used.

Finding correspondences The first step in any visual homing algorithm is matching the image currently seen by the robot with the target image specifying the target position. To this end, we extract local, affinely invariant regions from both images, and match these as explained in chapter 5. In this application, only the geometry-based regions are used, to limit the needed computation time.

Deriving the epipolar geometry From the corresponding regions between both images, we then derive the epipolar geometry in a robust way, using RANSAC [FB81] similar to the method used in chapter 7. However, this time we work with the essential matrix $\mathbf{E}$ instead of the fundamental matrix $\mathbf{F}$, as we need calibrated conditions for this application. This is not a problem though, since in most robot applications, the internal camera parameters are readily available.

The relation between any point $\mathtt{x}_1$ in the first image and its corresponding point $\mathtt{x}_2$ in the second image (after conversion to normalized coordinates $\mathtt{x}'_1$ and $\mathtt{x}'_2$ using the internal camera parameter matrices $\mathbf{K_1}$ and $\mathbf{K_2}$) is then given by:

$$\mathtt{x}_2'^{\mathtt{T}}\,\mathbf{E}\,\mathtt{x}'_1 = 0 \qquad with \quad \mathtt{x}'_\mathtt{i} = \mathbf{K_i}^{-1}\,\mathtt{x}_\mathtt{i} \qquad i = 1,2 \tag{10.1}$$

As long as we do not have a degenerate case (as will happen, for example, if all the correspondences lie on a plane in 3D, or if the motion between the two cameras is a pure rotation), the relative rotation and the direction of the relative translation can be determined directly from the essential matrix $\mathbf{E}$ [Fau93, LH81]. It is important to detect such degenerate cases and handle them appropriately. This can be done for instance by looking at the singular values of the set of linear equations expressing the relation between corresponding point coordinates and $\mathbf{E}$. We refer to [TFZ98] for a more detailed discussion of this subject.

Relative rotation To derive the relative rotation from the essential matrix, we refer to the work of Maybank [May93]. There, it is explained that the relative rotation $\mathbf{R}$ can be computed based on the singular value decomposition of $\mathbf{E}$, as follows:

$$\mathbf{R} = \mathbf{U}\,\mathbf{R}_{\frac{\pi}{2}}\,\mathbf{V}^T \quad with \quad \mathbf{E} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T \tag{10.2}$$

with $\mathbf{U}$ and $\mathbf{V}$ the matrices containing the left and right singular vectors of $\mathbf{E}$, and $\mathbf{R}_{\frac{\pi}{2}}$ a matrix representing a rotation around the Z axis over $\frac{\pi}{2}$ or $-\frac{\pi}{2}$. Evidently, this will give two different solutions (one for $\frac{\pi}{2}$ and one for $-\frac{\pi}{2}$), both satisfying the epipolar geometry. The one for which the change in direction of the optical camera axis is the smallest is the correct one, since we must be looking to the object from the same side.

In the following, we will try to give a geometric interpretation of Equation 10.2, that explains where the mysterious rotation over 90 degrees comes

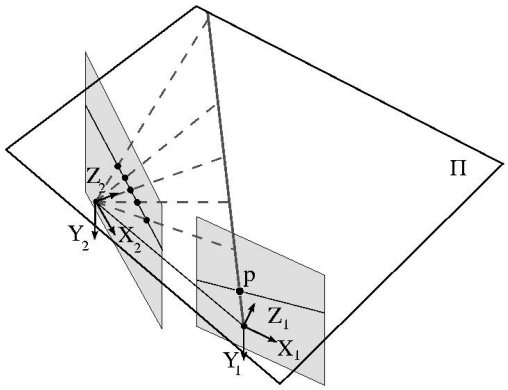


Figure 10.1: *Epipolar geometry revisited*

from, by interpreting the epipolar geometry in 3D (see Figure 10.1). Normally, $\mathtt{x}'_i$ is interpreted as the (metric) homogeneous coordinates of a 2D point in the image plane of the i-th camera. The relation between $\mathtt{x}'_1$ and $\mathtt{x}'_2$ given by the epipolar constraint (Equation 10.1) then states that the point $\mathtt{x}'_2$ should lie on the epipolar line with parameters given by $\mathbf{E}\mathtt{x}'_1$. However, we can also interpret the same entities directly in 3D. In that case, $\mathtt{x}'_i$ represents a *line* through the origin, expressed in the coordinate axes frame fixed to the i-th camera. The epipolar constraint in Equation 10.1 then states that the line $\mathtt{x}'_2$ should lie in the epipolar plane with parameters given by $\mathbf{E}\mathtt{x}'_1$. Now, we know that the parameters of a plane in 3D correspond to the coordinates of the normal to that plane. So the effect of $\mathbf{E}$ on $\mathtt{x}'_1$ is a transformation that maps all lines in a specific epipolar plane (expressed in the coordinate axes frame fixed to the first camera) to the normal on the same epipolar plane, but now expressed in the coordinate axes frame fixed to the second camera.

The transformation $\mathbf{E}$ can then be decomposed in a sequence of coordinate transformations, a projection, a translation and a rotation (starting from the coordinate axes frame fixed to the first camera) as follows (with each step illustrated in Figure 10.2):

1. a rotation of the coordinate axes frame that aligns the Z-axis with the join between the two optical centers of the two cameras (such that all epipolar planes intersect along the Z-axis) (from Figure 10.2(**a**) to (**b**)),
2. a projection of the line $\mathtt{x}'$ onto the XY-plane (see Figure 10.2(**c**)),
3. a translation of both the line $\mathtt{x}'$ and the coordinate axes frame along the Z-axis to the optical center of the second camera (see Figure 10.2(**d**)),
4. a rotation of the line $\mathtt{x}'$ around the Z-axis of 90 degrees (i.e. switching from a line in the plane to the normal to the plane) (see Figure 10.2(**e**)) and

5. a rotation of the coordinate axes frame bringing the Z-axis perpendicular to the image plane of the second camera, i.e. in alignment with the coordinate axes frame fixed to the second camera (from Figure 10.2(**e**) to (**f**)).

These are exactly the components of the singular value decomposition of $\mathbf{E}$, with $\mathbf{U}$ the first rotation of the coordinate axes frame (step 1), $\mathbf{\Sigma}$ the projection onto the XY-plane (step 2), and $\mathbf{V}$ the combination of the last two rotations (step 4 and 5). The translation in step 3 can be omitted, as this has no net effect on the equations, since both the line $\mathtt{x}'$ and the coordinate axes frame are translated in the same way.

To find the relative rotation between the two cameras, we only need the combination of steps (1) and (5), so we leave out the projection $\mathbf{\Sigma}$ (step 2), and add an extra rotation of 90 degrees $\mathbf{R}_{\frac{\pi}{2}}$ to compensate for step 4. This gives us exactly the formula given in Equation 10.2.

Direction of the relative translation The direction of the relative translation corresponds to the position of the epipole $\mathtt{e}$ (expressed in homogeneous coordinates) (see Figure 10.1), which, in turn, satisfies the following equation

$$\mathtt{t} = \mathtt{e} \qquad \textit{and} \qquad \mathbf{E}\mathtt{e} = 0 \tag{10.3}$$

So $\mathtt{e}$ can be found through an eigenanalysis of the essential matrix $\mathbf{E}$ (or, alternatively, from the matrix $\mathbf{U}$ in Equation 10.2).

Magnitude of the relative translation The magnitude of the relative translation cannot be derived from the epipolar geometry. This is only possible after taking a third image. Here, we follow the strategy as developed by Basri *et al.* [BasAl98, BasAl99]. To facilitate the computations, the three images are chosen in a special way, namely such that the relative motion between them is reduced to a pure translation (twice in the same direction).

To this end, we rotate the camera (or the robot) such that the image plane becomes parallel to the image plane of the target image (using the relative rotation computed earlier) and take a first image. Then, the robot is moved one step (for instance 20 cm) in the direction of the relative translation and a second image is taken. The third image corresponds to the target image itself.

Under these conditions the number of additional steps n towards the target image location is given by the cross-ratio of the point-coordinates in the three images ($\mathtt{x}_1$, $\mathtt{x}_2$ and $\mathtt{x}_t$) and the coordinates of the epipole ($\mathtt{x}_e$)(see [BasAl98]).

$$n = \frac{\|\mathbf{x_t} - \mathbf{x_2}\|.\|\mathbf{x_1} - \mathbf{x_e}\|}{\|\mathbf{x_2} - \mathbf{x_1}\|.\|\mathbf{x_t} - \mathbf{x_e}\|} \tag{10.4}$$

This can easily be understood once it is realized that the epipole corresponds to the vanishing point of the translation direction, and taking into account that the crossratio of four collinear points is a projective invariant, so remains unaltered under image projection.

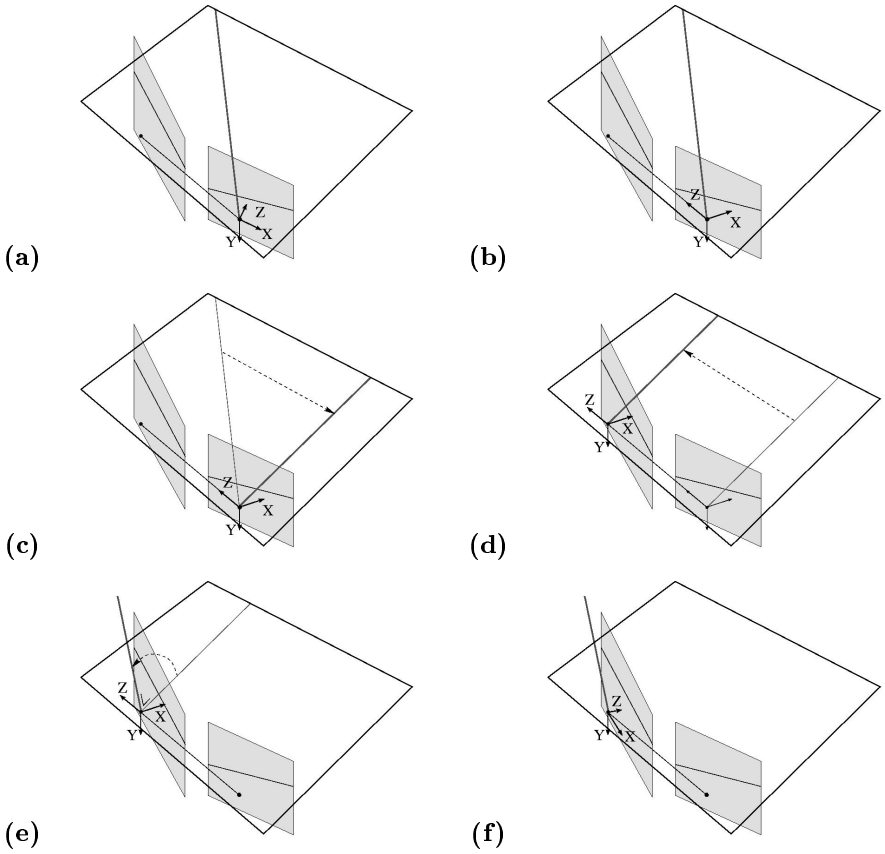


Figure 10.2: *The effect of the essential matrix* $\mathbf{E}$ *on the metric homogeneous coordinates of a point can be interpreted geometrically in 3D as a series of simple geometric transformations that map the corresponding line through the origin (expressed in the first image's canonical coordinate axes frame) onto the normal to the epipolar plane (expressed in the second image's canonical coordinate axes frame): (1) a rotation of the coordinate axes frame, (2) a projection onto the* XY*-plane, (3) a translation of the line and of the coordinate axes frame (4) a rotation of 90 degrees and (5) a rotation of the coordinate axes frame.*



Figure 10.3: *Our mobile robot*

Implementation details

We implemented the above algorithms on our home-built mobile robot, shown in Figure 10.3, which has a color CCD video conferencing camera mounted on it, that can zoom, pan and tilt. However, in this application, the zoom and tilt are not used.

While the robot moves, the correspondences with the target image are tracked from one image to the next using the KLT feature tracker [ST94, TomAl98]. This seriously slows down the speed of our robot. Currently, we are working at some optimizations in this respect.

Since our robot is non-holonomic, it first has to orient itself in the direction of the translation. Then the relative rotation is compensated for using the pan of the camera mounted on top of the robot. Next, the robot is moved one step forward, a new image is taken and the magnitude of the relative translation is computed. If the distance is too large, the procedure is applied in an iterative way. Finally, an extra rotation-step can be added to restore the robots orientation (with respect to the camera) if the camera-angle while taking the target image has been stored.

Experimental results

Figure 10.4 shows some experimental results. Each row represents a different experiment. Each time, the image on the left is the image taken at the start position of the robot. The image in the middle is the target image, while the image on the right is an image taken after the visual homing procedure.

Taking into account the limited accuracy of our robot (e.g. the pan of the camera can only be adjusted in steps of one degree, and the robot rotations

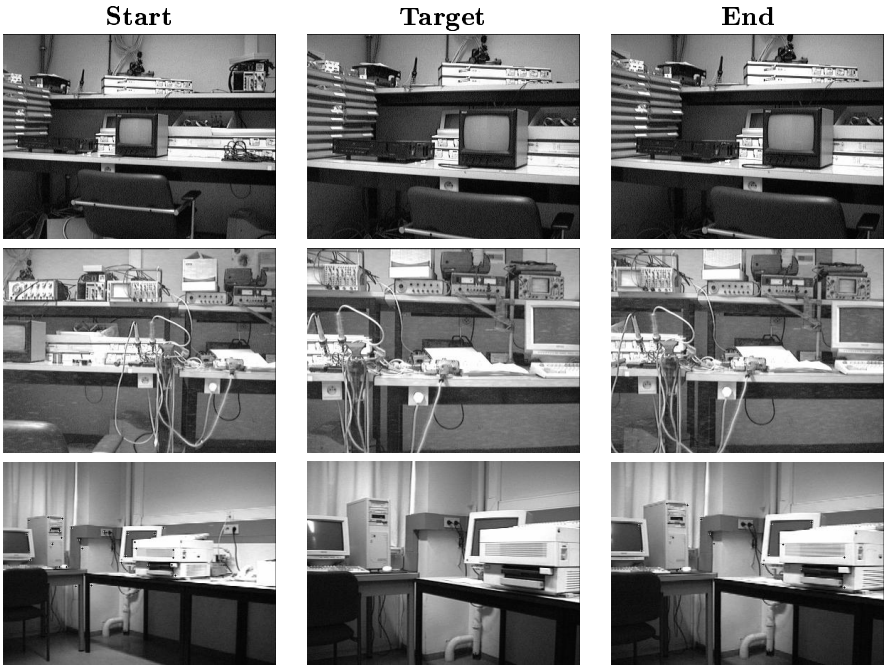


Figure 10.4: *Starting from a position corresponding to the image at the left and given the target image shown in the middle, the robot moves until it reaches the position corresponding to the image at the right.*

are probably even less accurate), the end images correspond very well to the respective target images.

In most of the examples shown in Figure 10.4, the motion between the start position and the target position consists mainly of a translation forward. The reason for this is of a practical nature: the camera mounted on our robot can only pan between -40 and 40 degrees, and our robot is non-holonomic. If the robot has to translate in another direction than straight forward, it first has to rotate until it faces this direction. However, the camera must be able to compensate for this rotation by panning in the opposite direction.

10.3 Vision-based mobile robot navigation

In general, one can distinguish two different approaches to mobile robot navigation: the *reconstructionist* approach and the *reactive* approach. In the more traditional reconstructionist approach, sensor information is used to construct a 3D model or metric map of the robot's environment (e.g. [KK93, KriAl93]). This approach requires much effort to maintain a consistent and accurate representation of the geometry of the world. On the other hand, the reactive

paradigm, initially proposed by Brooks [Bro86], bases the robots behavior more directly on immediate sensor data and less on a stored representation. In particular, large scale reconstruction is avoided, since it is argued that the world is not static, so it is hard to maintain a consistent representation, and perhaps most importantly, it is not needed for most navigation tasks either.

An important step in any reactive vision-based navigation system is *robot localization*, i.e. extract accurate information about the robot's current position. Carlsson [Car94] uses the view variation of the crossratio of four non-coplanar vertical lines, while Sim and Dudek [SD99] exploit the changes in the local appearance of small 'image landmarks'. However, the robot must learn which variation corresponds to which position. Hence these methods typically require an extensive learning phase.

In our method, on the other hand, it is not possible to obtain the exact robot position from a single image, as this position is only determined relative to a known reference position up to a translation in a known direction. However, it is always possible to move the robot to that reference position by applying the visual homing algorithm. In our opinion, the inconvenience due to this decrease in localization flexibility is by large compensated for by the fact that we do not need such an extensive learning phase, as our robot can generalize to a large extent from a single view, so a few reference views suffice.

In this section, two different strategies for mobile robot navigation based on local, affinely invariant regions are proposed. The first one is basically a sequential application of the visual homing algorithm developed in the previous section, while the second one uses the regions as natural landmarks or beacons.

Mobile robot navigation based on sequential visual homing The visual homing algorithm described in the previous section can be used as a basic building block for a complete vision-based navigation system.

First, during a learning phase, the robot is lead around the environment by a human operator. While moving around, the robot saves images at specific locations. Then, later on, in the execution phase, it can repeat the same path on its own, simply by visual homing through the sequence of images stored in the learning phase.

We can even go a step further. As explained in Rasmussen [RH96], a robot that can recognize places based on images taken at these locations can automatically build a topological map of his environment. The nodes in this map correspond to specific locations, characterized by one or more images as seen by the robot when it is at that location. The links between different nodes represent algorithms to navigate from one node to another. This can be a series of odometric data, collected during the learning phase; a wall-following algorithm based on ultrasonic sensing or – for a completely vision-based navigation – a visual homing algorithm and a series of target images. With such a topological map at hand, the robot can not only repeat a path learned by the human operator, but also combine parts of several paths to find the shortest way from one place to another.

During the learning phase, the robot collects images while moving around. From these, it selects a set of reference images that will be used for the visual homing in the execution phase. This can be done for instance by tracking some basic features in the images, and adding a new reference image each time the tracker has lost a certain percentage of these features. In this way, it is guaranteed that there is always a sufficient amount of overlap between two consecutive images in the visual homing sequence, while the amount of images that needs to be stored is kept within reasonable limits.

At the same time, the robot compares each new reference image with the already collected reference images up to that point to detect places which it has already visited before, as this would mean that the robot has to close a loop in his topological map.

Thanks to the invariance to viewpoint and illumination changes of the local, affinely invariant regions, only a small amount of reference images has to be stored for the visual homing sequence and a place can be recognized by the robot even if it is not exactly at the same position as the previous time.

Mobile robot navigation based on natural landmarks Alternatively, the local, affinely invariant regions can also be used as *natural landmarks* or *beacons* in an odometry-based navigation system. In this approach, the robot tries to build a metric map of his environment, based on information from wheel encoders and possibly an inertial measurement unit as well. The problem is that the information extracted from the encoders has errors that grow without bounds, so information from other sensors is required to keep the errors within reasonable limits.

A possible strategy goes as follows:

1. From the first image a small number of good local invariant regions are selected (based on uniqueness, discriminative power, stability, etc.). They are each uniquely identifiable based on their feature vector of moment invariants. From geometry we calculate the angles or *bearing* between the robot axis and the physical features corresponding to those regions.
2. After a short time interval, a new image is taken and the regions extracted from the previous image are tracked to the new image.
3. Next, the bearings from the robot to these features are computed again. and used by a Kalman filter to localize the vehicle and estimate the feature locations in 3D simultaneously.
4. Based on the updated robot location, and the estimated position of the features, it is checked whether any of the features that have gone out of sight or simply have been lost by the tracker, may be visible again from the current robot position. If so, new regions are extracted from the image in a region of interest around the estimated position of the region and compared with the stored invariant description of the feature we are looking for to see if we can pick it up again.

5. If needed, additional, new local invariant regions are added to the system, selected based on the same criteria as in step 1.
6. Go back to 2.

The nice thing of this strategy is that it allows *simultaneous localization and map building* or SLAM for short, i.e. no learning phase is needed as with the robot navigation algorithm based on successive application of the visual homing algorithm. The robot starts in an unknown location in an unknown environment and then incrementally builds a map of this environment while simultaneously using this map to compute absolute vehicle location.

Various SLAM systems have already been proposed, based on different forms of sensor information, including visual features (e.g. [BG97, NRT97, LM99, DM98]). Specifically, Davison and Murray [DM98] use small rectangular regions around Harris corner points as local features, and select new features in an intelligent way to keep the total number of features small. The advantage of using our local invariant regions in this context is that they can be detected from a wide range of viewpoints and still have sufficient discriminative power to tell them apart. The first point is important as this allows to work with a small number of features and will result in more accurate robot and feature localization. The second point is important, as the uncertainty of the robot location can increase dramatically if it moves in one direction for a long period, such that the uncertainty on the expected locations of the image features is too large and the position of the features alone is insufficient to match them.

10.4 Conclusion

In this chapter, we discussed the application of local invariant regions in the context of mobile robot navigation. First, an algorithm for visual homing based on the matching of regions and the analysis of the resulting epipolar geometry has been proposed. Next, two possible strategies for vision-based mobile robot navigation using the local invariant regions have been proposed. The first one relies on a sequential application of the visual homing algorithm while the second one uses the regions as natural landmarks or beacons.

Although the algorithms proposed in this chapter seem to hold good promise, the main bottleneck to date is the long computation time needed to extract the invariant regions. In some situations, this is acceptable (e.g. for robot localization after a power shutdown). However, in general this makes it hard to use them in a feedback-loop or during online computations.

Chapter 11

Conclusion

"Would you tell me, please, which way
I ought to go from here ?" said Alice.
"That depends for a good deal on
where you want to get to", said the Cat.
`Lewis Caroll`, *Alice in Wonderland.*

11.1 Summary

The starting point for this dissertation was the work of Schmid and Mohr [SM97], who combined the advantages of the model-based and appearance based approaches to object recognition, through the use of *local invariant features*. On the one hand, their approach inherits from the appearance-based school the ability to deal with objects of high complexity. On the other hand, their approach is less sensitive to scene clutter and occlusions and it can to a certain extent generalize from a single view, advantages of using local invariants.

We have extended this approach to invariance under *affine transformations*, by introducing a method to delineate parallelogram-shaped and elliptic regions that automatically adopt different shapes for different viewpoints, such that they systematically cover the same, physical part of a surface. The crux of the matter is that these corresponding shapes are determined solely on the basis of a single image, i.e. no other views are necessary. Corresponding regions are formed automatically in different images separately, without any knowledge about the other images. This can be seen as a generalization of the automatic scale detection proposed by Lindeberg [Lin98].

Once such invariant regions have been extracted, the color or intensity pattern they enclose is characterized with generalized color moment invariants. Both the region extraction and their invariant description are invariant under affine geometric changes and different scalings and offsets in the three color bands. The affine geometric invariance subsumes that the regions correspond

to (more or less) planar surface patches, which is often a reasonable assumption due to the local character of the regions. Such invariance implies that correspondences can be found under wider changes in viewpoint. For object recognition, this means that the number of reference images can be reduced.

Several region extraction methods have been proposed. The first one firmly relies on the extraction of geometric entities in the image, and is therefore coined *geometry-based region extraction*. The second method, on the other hand, directly exploits the image intensities, so we coined it the *intensity-based region extraction* method. The idea is to build an *opportunistic system*, that works with different types of regions, exploiting what's on offer in a specific image. In this way, the applicability of our system is increased, as well as the robustness. Other methods for extracting local, affinely invariant regions can still be added to our system, both general ones as more specific ones tuned towards specific features or applications. Based on the two methods proposed in this work, the feasibility of affine invariance on a local scale has been demonstrated.

No matter which method is used for finding correspondences between two views of an object or scene, there will always be a number of false matches at first. These can be due to symmetries in the image, or simply because the local region's distinctive power is not good enough. Using some new, semi-local constraints, the majority of these false matches can be rejected.

While finding correspondences and deriving the epipolar geometry can already be useful on its own, some more evolved applications have been developed as well. Experiments have shown that the matching of local affinely invariant regions can successfully be applied to *object recognition* and *object localization*. Moreover, the usefullness of such local features in the context of *content-based image retrieval* has been demonstrated as well. Finally, they are at the basis of a *visual homing* algorithm for an intelligent vision-based mobile robot. Apart from these realizations, many more applications are possible. Here, we think of vision-based robot navigation, image mosaicing, automatic video annotation or grouping, to name just a few.

11.2 Discussion and further research

The affinely invariant region extraction and matching described in this work can still be improved in a number of ways. First of all, we did not pay much attention to *efficiency* issues yet. Our main goal was to test the feasibility of affine photometric invariants on a local scale. So we restricted ourselves to the development of a system that illustrates the potential, without necessarily being optimal. Nevertheless, for many applications, processing speed is an important issue. Content-based image retrieval, for instance, is mostly interactive, with a user impatiently waiting for the results to appear on the screen of his computer. In mobile robot applications, processing speed is even more important. As a robot needs to interact with and react to its environment, it should be able to quickly draw its conclusions from a given input image. The algorithms

proposed in this work could be coded more efficiently. Nevertheless, this won't suffice to make them work in realtime. Further research is needed to find ways to increase the efficiency without compromising the generality of our approach.

Also, the stability, uniqueness, discriminative power and robustness of individual regions could be further investigated. Here we think of adding a *quality label* to each region. Not all regions found in one view are found in another view as well, and even if they are, they are not always matched correctly. The quality label should give a prediction of the probability of correctly matching the region. Such label could be based on statistical information about succesful and unsuccesful regions. For instance, homogeneous regions are known to be less disriminative, as there are typically many similar regions they can be matched to. Also information gathered during the region extraction could be used in this respect. For instance, one can get an idea of the robustness to noise of a region based on whether the extrema used were clearly outspoken or not. Finally, in some applications, a region can be tracked over several images, such that the quality of a region can be learnt from its history. This could be used for instance to reject non-planar and unstable regions. Such quality labels allow to focus on the good regions first, while some very poor regions can immediately be discarded. In this way, both the robustness and the efficiency can be improved. To use these quality labels to the fullest extent, a *probabilistic matching* scheme should be implemented instead of the simple voting mechanism used in this work. This also allows to speed up the process (e.g. in the context of content-based image retrieval), as not all regions in the query image have to be processed before one can draw conclusions.

We also plan to focus more specifically on the *object recognition*. In this context, the *automatic selection of reference views* is still an open issue. Given several views of an object that densely sample the entire viewing sphere, how can we automatically select a good subset to be used as reference views, that is at the same time minimal and complete ?

Alternatively, it is worth investigating whether there is no better representation than a set of reference views. A more *object-centered representation*, combining the local invariant regions from all reference views seems to hold good promise. Now, the votes are often spread over different reference views, as they always partially overlap. An object-centered approach would not have this kind of problems. Also, this approach would allow to gradually learn a model over time as more views become available (e.g. in a video sequence), without having to add new reference views all the time. How to keep the semi-local constraints in, and how to check the common visibility of different regions without reconstructing the whole object in 3D is still to be determined though.

Also interesting is the generalization from object recognition to *object classification* or *-categorisation*. Is it possible to extract some local, invariant features that are common to a whole class of objects such as faces or cars ? Although the local color or intensity patterns of individual objects belonging to the same class may differ a lot, some characteristics might prove more generic. A good example is symmetry, which is shared by all eyes over different faces or all

wheels over different cars. The automatic detection and extraction of such common features given a few example images of a specific class is still an open research domain.

Another field of research that could benefit from the matching of local invariant regions is *grouping*. This means combining bits and pieces of visual information into larger, semantically meaningful entities. Now, our system has serious problems if the image contains some regularity, such as a symmetry or repetition (e.g. several windows of the same type on a facade, or several identical tiles on a floor), as this results in a large number of ambiguous matches. However, the same matching technique could be used to detect these regularities, such that they can be exploited during the matching or recognition. More specifically, repeated patterns can be detected by clustering the local invariant regions based on their invariant descriptions. The matches and clusters found can then be used as input for a cascaded Hough transform [TPVG97], which allows to detect non-accidental alignments such as lines of fixed points or pencils of fixed lines. These fixed structures define subgroups of the general group of projectivities, leading to a significant reduction in the complexity of the problem. Some first preliminary results in this context will be published in [TTMVG01].

Bibliography

[AH93] S. Atiya and G. Hager, "Real-time Vision-based Robot Localization", *IEEE Trans. on Robotics and Automation*, 9(6):785-800, 1993.

[BahAl97] A. Bahi, R. Chaine, S. Bouakaz and D. Van Dorpe, "A Review on Aspect Graph Techniques", *Proc. IAPR-TC15 workshop on Graph Based Representations*, pp. 84-89, 1997.

[BW93] D. Ballard and L. Wixson, "Object Recognition using Steerable Filters at Multiple Scales", *IEEE Workshop on Qualitative Vision*, pp.2-10, 1993.

[BG98] C. Ballester and M. Gonzales, "Affine Invariant Texture Segmentation and Shape from Texture by Variational Methods", *Journal of Mathematical Imaging and Vision*, 9:141-171, 1998.

[BR95] R. Basri and E. Rivlin, "Localization and Homing using Combinations of Model Views", *Artificial Intelligence*, 78:327-354, 1995.

[BasAl98] R. Basri, E. Rivlin and I. Shimshoni, "Visual Homing: Surfing the Epipoles", *Proc. Int'l Conf. on Computer Vision*, pp.863-869, 1998.

[BasAl99] R. Basri, E. Rivlin and I. Shimshoni, "Image-Based Navigation under the Perspective Model", *Proc. Int'l Conf. on Robotics and Automation*, pp.2578-2583, 1999.

[Bed95] B. Bederson, "Audio Augmented Reality: A prototype Automated Tour Guide", *Proc. ACM Human Factors in Computing Systems*, pp.210-211, 1995.

[BG97] M. Betke and L. Gurvits, "Mobile Robot Localization using Landmarks", *IEEE Transactions on Robotics and Automation*, 1997.

[BM98] R. Behringer and N. Muller, "Autonomous Road Vehicle Guidance from Autobahnen to Narrow Curves", *IEEE Trans. Robotics and Automation*, 14(5):810-815, 1998.

[Bie92] I. Biederman, "From Image Edges to Geons to Viewpoint-Invariant Object Models: a Neural Net Implementation", *Proc. SPIE*, pp.570-578, 1992.

[BL93] T.O. Binford and T.S. Levitt, "Quasi-invariants: Theory and exploitation", *Proc. DARPA Image Understanding Workshop*, pp.819-829, 1993.

[BriAl92] M.H. Brill, E.B. Barrett and P.M. Payton, "Projective invariants for curves in two and three dimensions.", in: J. Mundy and A. Zisserman, eds., *Geometric Invariance in Computer Vision.* MIT Press, Chapter 9: 193-214, 1992.

[Bro86] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, 2(1):14-23, 1986.

[BWR90] J. Burns, R. Weiss and E. Riseman, "View Variation of Point Set and Line Segment Features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):51-68, 1990.

[DWR] California Departement of Water Resources database, `http://elib.cs.berkeley.edu/photos/dwr/`

[Can86] J. Canny " A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679-698, 1986.

[Car94] S. Carlsson, "Relative Positioning from Model Indexing", *Image and Vision Computing*, 12(3):179-186, 1994.

[Car00] S. Carlsson, "Recognizing Walking People", *Proc. European Conf. on Computer Vision*, 2000.

[CJ91] D.T. Clemens and D. W. Jacobs, "Space and Time Bounds on Indexing 3-D Models from 2-D Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007-1017, 1991.

[CCVC99] O. Chomat, V. Colin de Verdière and J.L. Crowley, "Recognizing Goldfish ? or Local Scale Selection for Recognition Techniques", *Proc. Symposium for Intelligent Robotics Systems*, 1999.

[CVC98] V. Colin de Verdière and J.L. Crowley, "Visual Recognition using Local Appearance", *Proc. European Conf. on Computer Vision*, pp.640-654, 1998.

[Cor96] P.I. Corke, "Visual Control of Robots: High-Performance Visual Servoing", Mechatronics, Research Studies Press (John Wiley), 1996.

[CygAl87] D. Cyganski, J. Orr, T. Cott and R. Dodson, "Development, implementation, testing, and application of an affine transform invariant curvature function" *Proc. Int'l Conf. on Computer Vision*, pp.496-500, 1987.

[DM98] A.J. Davison and D.W. Murray, "Mobile Robot Navigation Using Active Vision", *Proc. European Conf. on Computer Vision*, 1998.

[DP96] A. Del Bimbo and P. Pala, "Effective Image Retrieval Using Deformable Templates", *Proc. Int'l Conf. on Pattern Recognition*, pp.120-123, 1996.

[DP92] S.J. Dickinson and A.P. Pentland, "A Unified Approach to the Recognition of Expected and Unexpected Geon-based Objects", *Applications of Artificial Intelligence X: Machine Vision and Robotics*, vol. 1708, SPIE, pp.614-627, 1992.

[DZ96] G. Dudek and C. Zhang, "Vision-based Robot Localization without Explicit Object Models", *Proc. Int'l Conf. on Robotics and Automation*, pp.76-82, 1996.

[EM95] F. Ennesser and G. Medioni, "Finding Waldo, or Focus of Attention using Local Color Information", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):805-809, 1995.

[Fau92] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig", *Proc. European Conf. on Computer Vision*, pp.563-578, 1992.

[Fau93] O. Faugeras, "Three-Dimensional Computer Vision - a geometric viewpoint" MIT-Press, 1993.

[Fis38] R.A. Fischer, "The Statistical Utilization of Multiple Measurements", *Annals of Eugenics*, 8:376-386, 1938.

[FB81] M. Fischler and R. Bolles, "RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography", *Commun. Assoc. Comp. Mach.*, 24:381-95, 1981.

[QBIC95] M. Flickner *et al.*, "Query by Image and Video Content: the QBIC System", *IEEE Computer*, 28(9):23-32, 1995.

[FA91] W.T. Freeman and E.H. Adelson, "The Design and Use of Steerable Filters", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 13(9):891-906, 1991.

[Fri89] J.H. Friedman, "Regularized Discriminant Analysis", *Journal of the American Statistical Association*, 84:165-175, 1989.

[FF95] B.V. Funt and G.D. Finlayson, "Color Constant Color Indexing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):522-529, 1995.

[FBM98] B.V. Funt, K. Barnard and L. Martin, "Is Machine Colour Constancy Good Enough ?", *Proc. European Conf. on Computer Vision*, pp.445-459, 1998.

[GL94] J. Gårding and T. Lindeberg, "Shape-adapted Smoothing in Estimation of 3D Depth Cues from Affine Distortions of Local, 2D Brightness Structure", *Proc. European Conf. on Computer Vision*, pp.389-400, 1994.

[GL96] J. Gårding and T. Lindeberg, "Direct Computation of Shape Cues using Scale-adapted Spatial Derivative Operators", *Int'l Journal of Computer Vision*, 17(2):163-191, 1996.

[GPK98] N. Georgis, M. Petrou, J. Kittler "On the correspondence problem for wide angular separation of non-coplanar points", *Image and Vision Computing*, 16:35-41, 1998.

[Virage97] A. Gupta, "Visual Information Retrieval Technology: A Virage Perspective", Virage Technical Report, 1997.

[HCVC00] D. Hall, V. Colin de Verdière and J.L. Crowley, "Object Recognition using Colour Receptive Fields", *Proc. European Conf. on Computer Vision*, 2000.

[HanAl91] P. Hancock, R. Baddeley and L. Smith, "The Principal Components of Natural Images", *Computation in Neural Systems*, 3:61-70, 1991.

[Har88] C. Harris and M. Stephens, "A combined corner and edge detector", *Fourth Alvey Vision Conference*, pp.147-151, 1988.

[Har92] R. Hartley, "Estimation of Relative Camera Positions for Uncalibrated Cameras", *Proc. European Conf. on Computer Vision*, pp.579-587, 1992.

[Has93] K. Hashimoto, ed., "Visual Servoing", *Robotics and Automated Systems*, vol.7, World Scientific, 1993.

[HS94] G. Healey and D. Slater, "Using Illumination Invariant Color Histogram Descriptors for Recognition", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.355-360, 1994.

[Hu62] M.K. Hu, "Visual Pattern Recognition by Moment Invariants", *IRE Trans. Information Theory*, 8:179-187, 1962.

[HuaAl99] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu and R. Zabih, "Spatial Color Indexing and Applicaitions", *Int'l Journal of Computer Vision*, 35(3):245-268, 1999.

[Jac97] D.W. Jacobs, "Matching 3D Models to 2D Images", *Int'l Journal of Computer Vision*, 21(1/2):123-153, 1997.

[KSK88] G. Klinker, S.A. Shafer, T. Kanade, "The Measurement of Highlights in Color Images", *Int'l Journal of Computer Vision*, 2(1):7-32, 1988.

[KVD79] J.J. Koenderink and A. Van Doorn, "The internal representation of solid shape with respect to vision", *Biological Cybernetics*, 32:211-216, 1979.

[KVD87] J.J. Koenderink and A. Van Doorn, "Representation of Local Geometry in the Visual System", *Biological Cybernetics*, 55:367-375, 1987.

[KK93] A. Kosaka and A. Kak, "Fast Vision-Guided Mobile Robot Navigation using Model-based Reasoning and Prediction of Uncertainties", *CVGIP: Image Understanding*, 56(3):271-329, 1993.

[KriAl93] D.J. Kriegman, E. Triendl and T.O. Binford, "Stereo Vision and Navigation in Buildings for Mobile Robots", *IEEE Trans. on Robotics and Automation*, 5(6):792-803, 1989.

[Kru97] J. Krumm, "Object Detection with Vector Quantized Binary Features", *Proc. Int'l Conference on Computer Vision and Pattern Recognition*, pp.179-185, 1997.

[LW88a] Y. Lamdan and H. Wolfson, "Geometric Hashing: A general and efficient model based recognition scheme", *Proc. Int'l Conf. on Computer Vision*, pp.238-249, 1988.

[LW88b] Y. Lamdan, J. Schwartz and H. Wolfson, "Object Recognition by Affine Invariant Matching", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.335-344, 1988.

[Lin93] T. Lindeberg, "On Scale Selection for Differential Operators", *Proc. Scandinavian Conf. on Image Analysis*, pp.857-866, 1993.

[Lin98] T. Lindeberg, "Feature Detection with Automatic Scale Detection", *Int'l Journal of Computer Vision*, 30(2):79-116, 1998.

[LM99] S. Livatino and C. Madsen, "Optimization of Robot Self-Localization Accuracy by Automatic Visual-landmark Selection", *Proc. Scandinavian Conf. on Image Analysis*, pp.501-506, 1999.

[LP95] F. Liu and R.W. Picard, "Periodicity, Directionality, and Randomness: Wold Features for Image Modeling and Retrieval", M.I.T. Media Lab Perceptual Computing Section Technical report, No.320, 1995.

[LH81] H.C. Longuett-Higgins "A computer Algorithm for reconstructing a scene from two projections", *Nature*, 293:133-135, 1981.

[Low87] D.G. Lowe, "Three-dimensional Object Recognition from Single Two-dimensional Images", *Artificial Intelligence*, 31:355-395, 1987.

[Low99] D. Lowe, "Object Recognition from Local Scale-Invariant Features", *Proc. Int'l Conf. on Computer Vision*, pp.1150-1157, 1999.

[LF96] Q. T. Luong and O. D. Faugeras, "The Fundamental Matrix: theory, algorithms, and stability analysis", *Int'l Journal of Computer Vision*, 17(1):43-75, 1996.

[MMK95] J. Matas, R. Marik and J. Kittler, "On Representation and Matching of Multicolored Objects", *Proc. Int'l Conf. on Computer Vision*, pp.726-732, 1995.

[May93] S. Maybank, "Theory of Reconstruction from Image Motion", Springer Verlag, 1993.

[MRL96] D.P. McReynolds and D.G. Lowe, "Rigidity Checking of 3D Point Correspondences Under Perspective Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1174-1185, 1996.

[Mel97] B. Mel, "SEEMORE: Combining Color, Shape and Texture Histogramming in a Neurally-Inspired Approach to Visual Object Recognition", *Neural Computation*, 9:777-804, 1997.

[MMVG98] F. Mindru, T. Moons and L. Van Gool, "Color-based moment invariants for viewpoint and illumination independent recognition of planar color patterns", *Proc. Int'l Conf. on Advances in Pattern Recognition*, pp.113-122, 1998.

[MMVG99a] F. Mindru, T. Moons and L. Van Gool, "Recognizing Color Patterns Irrespective of Viewpoint and Illumination", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.368-373, 1999.

[MMVG99b] F. Mindru, T. Moons and L. Van Gool, "Viewpoint and Illumination Independent Recognition of Color Patterns", K.U.Leuven Techn. report KUL/ESAT/PSI/9901, 1999.

[MAK96] F. Mokhtarian, S. Abbasi and J. Kittler, "Robust and Efficient Shape Indexing through Curvature Scale Space", *Proc. British Machine Vision Conference*, 1996.

[MooAl95] T. Moons, E. Pauwels, L. Van Gool, P. Kempenaers, A. Oosterlinck, "Foundations of Semi-Differential Invariants", *Int'l Journal of Computer Vision*, 14(1):25-47, 1995.

[MZ92] J.L. Mundy, A. Zisserman and D. Forsyth, eds. *Geometric Invariance in Computer Vision*, MIT Press, 1992.

[MZ94] J.L. Mundy and A. Zisserman, eds. *Applications of Invariance in Computer Vision*, LNCS 825, Springer, 1994.

[MN95] H. Murase and S.K. Nayar, "Visual Learning and Recognition of 3D objects from appearance", *Int'l Journal of Computer Vision*, 14:5-24, 1995.

[NG98] K. Nagao and W.E.L. Grimson, "Using Photometric Invariants for 3D Object Recognition", *Computer Vision and Image Understanding*, 71(1):74-93, 1998.

[NB96] S.K. Nayar and R.M. Bolle, "Reflectance Based Object Recognition", *Int'l Journal of Computer Vision*, 17(3):219-240, 1996.

[NRT97] J. Neira, M.I. Ribeira and J.D. Tardos, "Mobile Robot Localization and Map Building using Monocular Vision", *Proc. of Symposium on Intelligent Robotics Systems*, 1997.

[OI96] K. Ohba and K. Ikeuchi, "Recognition of the Multi Specularity Objects for Bin-picking Task" *Proc. Int'l Conference on Inteligent Robotic Systems*, pp.1440-1448, 1996.

[OI97] K. Ohba and K. Ikeuchi, "Detectability, Uniqueness and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):1043-1048, 1997.

[ON94] M. Oren and S. Nayar, "Seeing Beyond Lambert's Law", *Proc. European Conf. on Computer Vision*, pp.269-280, 1994.

[PZ96] G. Pass and R. Zabih, "Histogram Refinement for Content-Based Image Retrieval", *IEEE Workshop on Applications of Computer Vision*, pp.96-102, 1996.

[PPS96] A. Pentland, R. Picard and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases", *Int'l Journal of Computer Vision*, 18(3):233-254, 1996.

[Pic95] R.W. Picard, "Light-years from Lena: Video and Image Libraries of the Future", MIT Technical Report no.339, 1995.

[Pop94] A.R. Pope, "Model-Based Object Recognition, A Survey of Recent Research", Technical Report, University of Columbia, 1994.

[PZ98a] P. Pritchett and A. Zisserman, "Wide Baseline Stereo Matching", *Proc. International Conference on Computer Vision*, Narosa Publishing House, pp.754-760, 1998.

[PZ98b] P. Pritchett and A. Zisserman, "Matching and Reconstruction from Widely Separate Views", *Proc. SMILE Workshop (post-ECCV'98)*, Lecture Notes in Computer Science, Vol. 1506, Springer-Verlag, pp.78-92, 1998.

[RB95] R. Rao and D. Ballard, "An Active Vision Architecture based on Iconic Representations", *Artificial Intelligence*, 78:461-505, 1995.

[RH96] C. Rasmussen and G. Hager, "Robot Navigation Using Image Sequences", *Proc. National Conference on Artificial Intelligence*, pp.938-943, 1996.

[Rei93] T. Reiss, "Recognizing Planar Objects using Invariant Image Features", *LNCS* 676, Springer, 1993.

[RH93] I. Rigoutsos and R. Hummel, "Distributed Bayesian Object Recognition", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.180-186, 1993.

[RocAl98] A. Roche, G. Malandain, X. Pennec and N. Ayache, "Multimodal Image Registration by Maximization of the Correlation Ratio", INRIA Technical Report nr.3378, 1998.

[RMA99] A. Roche, G. Malandain and N. Ayache, "Unifying Maximum Likelihood Approaches in Medical Image Registration", INRIA Technical Report nr.3741, 1999.

[Rou87] P.J.Rousseeuw "Robust Regression and Outlier Detection", Wiley, New York, 1987.

[SC96] B. Schiele and J.L. Crowley, "Object Recognition using Multidimensional Receptive Field Histograms", *Proc. European Conf. on Computer Vision*, pp.610-619, 1996.

[SchAl99] B. Schiele, N. Oliver, T. Jebara and A. Pentland, "An interactive Computer Vision System - DyPERS: Dynamic Personal Enhanced Reality System", *Proc. int'l Conf. on Vision Systems*, 1999.

[SM97] C. Schmid and R. Mohr, "Local Greyvalue Invariants for Image Retrieval", *IEEE transactions on Pattern Analysis and Machine Intelligence*, 19(5):872-877, may 1997.

[Sch99] C. Schmid, "A Structured Probabilistic Model for Recognition", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.485-490, 1999.

[SM00] C. Schmid, R. Mohr and C. Bauckhage, "Evaluation of Interest Point Detectors", *Int'l Journal of Computer Vision*, 37(2), 151-172, 2000.

[Sha96] A. Shashua, "The Rank 4 Constraint in Multiple View Geometry", *Proc. European Conf. on Computer Vision*, pp.196-206, 1996.

[ST94] J. Shi and C. Tomasi, "Good Features to Track", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.593-600, 1994.

[SB98] S. Siggelkow and H. Burkhardt, "Image Retrieval based on Local Invariant Features", *Proc. IASTED Int'l Conf. on Signal and Image Processing*, pp1998.

[SD99] R. Sim and G. Dudek, "Learning Visual Landmarks for Pose Estimation", *Proc. Int'l Conf. on Robotics and Automation*, pp.1972-1978, 1999.

[SCR95] D. Sinclair, H. Christensen and C. Rothwell, "Using the Relation between a Plane Projectivity and the Fundamental Matrix", *Proc. SCIA*, pp.181-188, 1995.

[SH95] D. Slater and G. Healey, "Combining Color and Geometric Information for the Illumination Invariant Recognition of 3D Objects", *Proc. Int'l Conf. on Computer Vision*, pp.563-568, 1995.

[SH96] D. Slater and G. Healey, "The Illumination Invariant Recognition of 3D Objects Using Local Color Invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):206-210, 1996.

[SRP95] S. Startchick, C. Rauber and T. Pun, "Recognition of Planar Objects over Complex Backgrounds using Line Invariants and Relevance Measures", *Workshop on Geometric Modeling and Invariants for Computer Vision*, pp.301-307, 1995.

[SMP98] S. Startchick, R. Milanese and T. Pun, "Projective and illumination invariant representation of disjoint shape", *Image and Vision Computing*, 16(9-10):713-723, 1998.

[Sug88] K. Sugihara, "Some Location Problems for Robot Navigation using a Single Camera", *Computer Vision, Graphics and Image Processing*, 42:112-129, 1988.

[SK97] B.J. Super and W.N. Klarquist, "Patch Matching and Stereopsis in a General Stereo Viewing Geometry", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):247-253, 1997.

[SB91] M. Swain and D. Ballard, "Color Indexing", *Int'l Journal on Computer Vision*, 7(1):11-32, 1991.

[TJr] TargetJr, `http://www.targetjr.org/`.

[TC88] C.-H. Teh and R. Chin, "On image analysis by the methods of moments", *IEEE Transactions on Pattern Analysis and Machine Intelligence*,10(4):496-513, 1988.

[TC99] D. Tell and S. Carlsson, "View Based Visual Servoing Using Epipolar Geometry", *Proc. Scandinavian Conference on Image Analysis*, pp.63-70, 1999.

[TC00] D. Tell and S. Carlsson, "Wide Baseline Stereo Matching using Affine Invariants Computed from Intensity Profiles", *Proc. European Conf. on Computer Vision*, 2000.

[Tho90] C.E. Thorpe, "Vision and Navigation, the Carnegie Mellon NAVLAB", Kluwer, 1990.

[TomAl98] T. Tommasini, A. Fusiello, E. Trucco and V. Roberto, "Making Good Features to Track Better", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.189-183, 1998.

[TBM94] P. Torr, P. Beardsley and D. Murray, "Robust Vision", *Proc. British Machine Vision Conference*, 1994.

[Tor95] P. Torr, "Motion Segmentation and Outlier Detection", PhD Thesis, Dept. of Engineering Science, University of Oxford, 1995.

[TFZ98] P. Torr, A.W. Fitzgibbon and A. Zisserman, "Maintaining Multiple Motion Models Hypotheses over Many Views to Recover Matching and Structure", *Proc. Int'l Conf. on Computer Vision*, pp.485-491, 1998.

[TTMVG01] A. Turina, T. Tuytelaars, T. Moons and L. Van Gool, "Grouping via the Matching of Repeated Patterns", to appear in *Proc. Int'l Conf. on Advances in Pattern Recognition*, 2001.

[TPVG97] T. Tuytelaars, M. Proesmans and L. Van Gool, "The Cascaded Hough Transform as Support for Grouping and Finding Vanishing Lines and Points", *Proc. Int'l Workshop on Algebraic Frames for the Perception-Action Cycle*, Lecture Notes in Computer Science, vol. 1315, pp.278-289, Sommer G. and Koenderink J.J. eds., Springer-Verlag, 1997.

[TuyAl99a] T. Tuytelaars, L. Van Gool, L. D'haene and R. Koch, "Matching Affinely Invariant Regions for Visual Servoing", *Proc. Int'l Conf. on Robotics and Automation*, pp.1601-1606, 1999.

[TuyAl99b] T. Tuytelaars, M. Vergauwen, M. Pollefeys and L. Van Gool "Image Matching for Wide baseline Stereo", *Proc. Int'l Conf. on Forensic Human Identification in the Millenium*, pp.1601-1606, 1999.

[TVG99] T. Tuytelaars and L. Van Gool, "Content-based Image Retrieval based on Local Affinely Invariant Regions", *Proc. Int'l Conf. on Visual Information Systems*, pp.493-500, 1999.

[TuyAl00a] T. Tuytelaars, A. Zaatri, L. Van Gool and H. Van Brussel, "Automatic Object Recognition as part of an Integrated Supervisory Control System", *Proc. Int'l Conf. on Robotics and Automation*, pp.3707-3712, 2000.

[TVG00] T. Tuytelaars and L. Van Gool, "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions", *Proc. British Machine Vision Conference*, 2000.

[Ull89] S. Ullman, "Alligning Pictorial Descriptions: An Approach to Object Recognition", *Cognition*, 32:193-254, 1989.

[UB91] S. Ullman and R. Basri, "Recognition by Linear Combinations of Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:992-1006, 1991.

[VGTP99] L. Van Gool, T. Tuytelaars and M. Pollefeys, "Adventurous Tourism for Couch Potatoes", Lecture Notes in Computer Science, vol.1689, pp.98-107, Solina F. and Leonardis A., eds., Springer-Verlag 1999 (*Proc. of the Int'l Conf. on Computer Analysis of Images and Patterns*), 1999.

[VGTT01] L. Van Gool, T. Tuytelaars and A. Turina, "Local Features for Image Retrieval", to appear in *State-of-the-Art in Content-Based Video and Image Retrieval*, H. Burkhardt, H.-P. Kriegel, R. Veltkamp, eds., Kluwer Publishers, 2001.

[VGAl90] L. Van Gool, J. Wagemans, J. Vandeneede and A. Oosterlinck, "Similarity extraction and modelling", *Proc. Int'l Conf. on Computer Vision*, pp.530-534, 1990.

[Vio96] P. Viola, "Complex Feature Recognition: A Bayesian Approach for Learning to Recognize Objects", Technical Report MIT AI Lab 1591, 1996.

[VW97] P. Viola and W. Wells, "Alignment by Maximization of Mutual Information", *Int'l Journal of Computer Vision*, 24(2):137-154, 1997.

[Wei88] I. Weiss, "Projective invariants of shapes" *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.291–297, 1988.

[Wol94] L. Wolff, "On the Relative Brightness of Specular and Diffuse Reflection", *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pp.369-376, 1994.

[Wol90] H. Wolfson, "Model-based Object Recognition by Geometric Hashing", *Proc. European Conf. on Computer Vision*, pp.526-536, 1990.

[Woo96] J. Wood, "Invariant Pattern recognition: A Review", *Pattern Recognition*, 29(1):1-17, 1996.

[ZaaAl00] A. Zaatri, T. Tuytelaars, R. Waarsing, H. Van Brussel and L. Van Gool, "Design and implementation of a supervised intelligent function", *Proc. Int'l Conf. on Artificial Intelligence*, pp.799-805, 2000.

[ZVB99] A. Zaatri and H. Van Brussel, "Improvements and Performance Evaluation for Integrated Supervisory Control Systems", submitted to *Autonomous Systems Journal*, 1999.

[ZhaAl95] Z. Zhang, R. Deriche, O. Faugeras and Q.T. Luong, "A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry", *Artifical Intelligence Journal*, 78:87-119, 1995.

[ZisAl94] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu and N. Pillow, "3D Object Recognition using Invariance", University of Oxford Techn. Report OUEL 2027/94, 1994.

Appendix A

Affine invariance of function extrema

Suppose we have the following geometric and photometric deformations between two views:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} s_R & 0 & 0 \\ 0 & s_G & 0 \\ 0 & 0 & s_B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} t_R \\ t_G \\ t_B \end{bmatrix}$$

with (R, G, B) and (R', G', B') the three different colorbands and (x, y) and (x', y') the coordinates of corresponding points. In the sequel, we will use I and I' to refer to either of the three colorbands R, G or B.

We will now prove that the extrema of the functions $f_1(\Omega)$, $f_2(\Omega)$, $f_3(\Omega)$ and $f_I(t)$ given in Equations 4.5, 4.6, 4.7 and 4.9 are invariant to the above deformations. In other words, for each region Ω in image I for which $f(\Omega)$ reaches an extremum, there must be a corresponding region Ω' in image I' for which $f'(\Omega')$ reaches an extremum as well.

Affinely invariant extrema of $f_1(\Omega)$

As mentioned already in the text, the first function $f_1(\Omega)$ represents the average intensity over the region. The extrema of this function being invariant to the considered deformations, can easily be understood intuitively. Here, we give a

more formal proof.

$$\begin{aligned} f_1(\Omega) &= \frac{\int_\Omega I(x,y)\ dxdy}{\int_\Omega dxdy} \\ f_1'(\Omega') &= \frac{\int_{\Omega'} I'(x',y')\ dx'dy'}{\int_{\Omega'} dx'dy'} \\ &= \frac{\int_\Omega (sI(x,y)+t)\ (ad-bc)\ dxdy}{\int_\Omega (ad-bc)dxdy} \\ &= s\frac{\int_\Omega I(x,y)\ dxdy}{\int_\Omega dxdy} + t \\ &= sf_1(\Omega) + t \end{aligned}$$

Clearly, if s is positive,

$$f_1(\Omega_1) > f_1(\Omega_2) \iff f_1'(\Omega_1') > f_1'(\Omega_2')$$

Hence, extrema of the function $f_1(\Omega)$ are preserved under the considered deformations. If, on the other hand, s is negative, then

$$f_1(\Omega_1) > f_1(\Omega_2) \iff f_1'(\Omega_1') < f_1'(\Omega_2')$$

As a result, minima for the function $f_1(\Omega)$ become maxima for the function $f_1'(\Omega')$ and vice versa. Again, extrema are preserved. The case where $s = 0$ does not correspond to realistic photometric changes, and hence can be discarded in our analysis.

Effects of the deformations on the center of gravity

For the functions $f_2(\Omega)$ and $f_3(\Omega)$ of Equations 4.6 and 4.7, it is important to first fully understand the effect of the deformations on the center of gravity

$$\mathtt{p_g} = (x_g, y_g) = (\frac{\int_\Omega I(x,y)\ x\ dxdy}{\int_\Omega I(x,y)\ dxdy}, \frac{\int_\Omega I(x,y)\ y\ dxdy}{\int_\Omega I(x,y)\ dxdy})$$

First, let us consider only geometric deformations. In that case, we get

$$\begin{aligned} \mathtt{p_g'} &= (x_g', y_g') \\ x_g' &= \frac{\int_{\Omega'} I'(x',y')\ x'\ dx'dy'}{\int_{\Omega'} I'(x',y')\ dx'dy'} \\ &= \frac{\int_\Omega I(x,y)\ (ax+by+e)\ (ad-bc)dxdy}{\int_{\Omega'} I(x,y)\ (ad-bc)dxdy} \\ &= a\frac{\int_\Omega I(x,y)\ x\ dxdy}{\int_{\Omega'} I(x,y)\ dxdy} + b\frac{\int_\Omega I(x,y)\ y\ dxdy}{\int_{\Omega'} I(x,y)\ dxdy} + e \\ &= ax_g + by_g + e \end{aligned}$$

$$
\begin{aligned}
y'_g &= \frac{\int_{\Omega'} I'(x',y') \;\; y' \;\; dx'dy'}{\int_{\Omega'} I'(x',y') \;\; dx'dy'} \\
&= \frac{\int_{\Omega} I(x,y) \;\; (cx+dy+f) \;\; (ad-bc)dxdy}{\int_{\Omega'} I(x,y) \;\; (ad-bc)dxdy} \\
&= c\frac{\int_{\Omega} I(x,y) \;\; x \;\; dxdy}{\int_{\Omega'} I(x,y) \;\; dxdy} + d\frac{\int_{\Omega} I(x,y) \;\; y \;\; dxdy}{\int_{\Omega'} I(x,y) \;\; dxdy} + f \\
&= cx_g + dy_g + f
\end{aligned}
$$

Hence, the center of gravity behaves as a normal, physical point under the affine deformations.

Now, let us consider the effect of photometric deformations. Here, we investigate the coordinates of the center of gravity relative to the coordinates of the region center

$$\mathtt{p_c} = (x_c, y_c) = (\frac{\int_{\Omega} x \;\; dxdy}{\int_{\Omega} dxdy}, \frac{\int_{\Omega} y \;\; dxdy}{\int_{\Omega} dxdy})$$

It can be shown that the only influence of the photometric changes is a shift of the center of gravity towards the center of the region:

$$
\begin{aligned}
x'_g - x'_c &= \frac{\int_{\Omega'} I'(x',y') \;\; x' \;\; dx'dy'}{\int_{\Omega'} I'(x',y') \;\; dx'dy'} - x'_c \\
&= \frac{\int_{\Omega} (sI(x,y)+t) \;\; x \;\; dxdy}{\int_{\Omega} (sI(x,y)+t) \;\; dxdy} - x_c \\
&= s\frac{\int_{\Omega} I(x,y)x \;\; dxdy}{\int_{\Omega} (sI(x,y)+t) \;\; dxdy} + t\frac{\int_{\Omega} x \;\; dxdy}{\int_{\Omega} (sI(x,y)+t) \;\; dxdy} - x_c \\
&= sx_g\frac{\int_{\Omega} I(x,y) \;\; dxdy}{\int_{\Omega} (sI(x,y)+t) \;\; dxdy} + tx_c\frac{\int_{\Omega} dxdy}{\int_{\Omega} (sI(x,y)+t) \;\; dxdy} - x_c \\
&= x_g\frac{sM^1_{00}}{sM^1_{00}+tM^0_{00}} + x_c\left(\frac{tM^0_{00}}{sM^1_{00}+tM^0_{00}} - 1\right) \\
&= x_g\frac{sM^1_{00}}{sM^1_{00}+tM^0_{00}} - x_c\frac{sM^1_{00}}{sM^1_{00}+tM^0_{00}} \\
&= (x_g - x_c)\frac{sM^1_{00}}{sM^1_{00}+tM^0_{00}} \\
y'_g - y'_c &= \ldots = (y_g - y_c)\frac{sM^1_{00}}{sM^1_{00}+tM^0_{00}}
\end{aligned}
$$

Apart from the ratio of the offset t and the scalefactor s, this shift also depends on the zeroth and first order, zeroth degree moments and hence on the intensity profile of the region Ω. Clearly, if the offset t is equal to zero, there is no shift and the center of gravity stays unaltered.

Affinely invariant extrema of $f_2(\Omega)$ and $f_3(\Omega)$

$f_2(\Omega)$ and $f_3(\Omega)$ are both composed of two factors, a ratio of two areas, one of which depends on the center of gravity, and an expression of moments up to

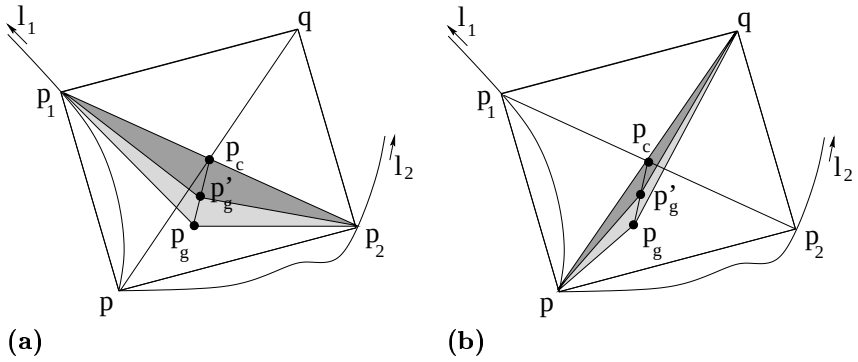


Figure A.1: *The effect of a photometric deformation on the first factor of* $f_2(\Omega)$ *(***a***) and* $f_3(\Omega)$ *(***b***).*

the second order.

$$
\begin{aligned}
f_2(\Omega) &= abs\left(\frac{\left|\mathtt{p}_1-\mathtt{p}_g \quad \mathtt{p}_2-\mathtt{p}_g\right|}{\left|\mathtt{p}-\mathtt{p}_1 \quad \mathtt{p}-\mathtt{p}_2\right|}\right)\frac{M^1_{00}}{\sqrt{M^2_{00}M^0_{00}-(M^1_{00})^2}} \\
f_3(\Omega) &= abs\left(\frac{\left|\mathtt{p}-\mathtt{p}_g \quad \mathtt{q}-\mathtt{p}_g\right|}{\left|\mathtt{p}-\mathtt{p}_1 \quad \mathtt{p}-\mathtt{p}_2\right|}\right)\frac{M^1_{00}}{\sqrt{M^2_{00}M^0_{00}-(M^1_{00})^2}} \\
&\textit{with} \quad M^n_{pq} = \int_\Omega I^n(x,y)x^p y^q \; dxdy \\
&\qquad \mathtt{p}_g = \left(\frac{M^1_{10}}{M^1_{00}}, \frac{M^1_{01}}{M^1_{00}}\right)
\end{aligned}
$$

The first factor is a ratio of two areas, defined by the points $\mathtt{p}$, $\mathtt{p}_1$ and $\mathtt{p}_2$ fixed to the region and the center of gravity $\mathtt{p}_g$. As we have seen in the previous section, the center of gravity behaves as a normal, physical point under the affine geometric deformations, such that this first factor clearly is affinely invariant.

Here, we show that the effect of the photometric deformations on this first factor is similar to their effect on the coordinates of the center of gravity relative to the region center, namely a resaling with the same scalefactor. This can be understood geometrically by the fact that the region center p_c lies on the diagonals of the parallelogram-shaped region, i.e. on the line connecting p and q on one hand and the line connecting p_1 and p_2 on the other hand, which also form one side of the areas used in the numerators of the first factors (see figure A.1). Hence the shift in the position of the center of gravity causes a proportional rescaling of the area:

$$\frac{\begin{vmatrix} \mathtt{p}' - \mathtt{p}'_g & \mathtt{q}' - \mathtt{p}'_g \end{vmatrix}}{\begin{vmatrix} \mathtt{p}' - \mathtt{p}'_1 & \mathtt{p}' - \mathtt{p}'_2 \end{vmatrix}} = \frac{\begin{vmatrix} \mathtt{p} - \mathtt{p}_g & \mathtt{q} - \mathtt{p}_g \end{vmatrix}}{\begin{vmatrix} \mathtt{p} - \mathtt{p}_1 & \mathtt{p} - \mathtt{p}_2 \end{vmatrix}} \frac{sM_{00}^1}{sM_{00}^1 + tM_{00}^0}$$

$$\frac{\begin{vmatrix} \mathtt{p}'_1 - \mathtt{p}'_g & \mathtt{p}'_2 - \mathtt{p}'_g \end{vmatrix}}{\begin{vmatrix} \mathtt{p}' - \mathtt{p}'_1 & \mathtt{p}' - \mathtt{p}'_2 \end{vmatrix}} = \frac{\begin{vmatrix} \mathtt{p}_1 - \mathtt{p}_g & \mathtt{p}_2 - \mathtt{p}_g \end{vmatrix}}{\begin{vmatrix} \mathtt{p} - \mathtt{p}_1 & \mathtt{p} - \mathtt{p}_2 \end{vmatrix}} \frac{sM_{00}^1}{sM_{00}^1 + tM_{00}^0}$$

This extra scalefactor must be compensated for by the second component in the expressions of $f_2(\Omega)$ and $f_3(\Omega)$. And indeed, the second component seems to have exactly the inverse scalefactor:

$$\begin{aligned}
\frac{M'^1_{00}}{\sqrt{M'^2_{00} M'^0_{00} - (M'^1_{00})^2}} &= \frac{\int_{\Omega'} I'(x',y')dx'dy'}{\sqrt{\left\{ \begin{array}{l} \int_{\Omega'} I'^2(x',y')dx'dy' \int_{\Omega'} dx'dy' \\ -\int_{\Omega'} I'(x',y')dx'dy' \int_{\Omega'} I'(x',y')dx'dy' \end{array} \right\}}} \\
&= \frac{\int_{\Omega} (sI(x,y)+t)(ad-bc)dxdy}{\sqrt{\left\{ \begin{array}{l} \int_{\Omega} (sI(x,y)+t)^2(ad-bc)dxdy \int_{\Omega} (ad-bc)dxdy \\ -\int_{\Omega} (sI(x,y)+t)(ad-bc)dxdy \int_{\Omega} (sI(x,y)+t)(ad-bc)dxdy \end{array} \right\}}} \\
&= \frac{sM_{00}^1 + tM_{00}^0}{\sqrt{(s^2 M_{00}^2 + 2stM_{00}^1 + t^2 M_{00}^0) M_{00}^0 - (sM_{00}^1 + tM_{00}^0)(sM_{00}^1 + tM_{00}^0)}} \\
&= \frac{sM_{00}^1 + tM_{00}^0}{\sqrt{\left\{ \begin{array}{l} s^2 M_{00}^2 M_{00}^0 + 2st M_{00}^1 M_{00}^0 + t^2 M_{00}^0 M_{00}^0 \\ -s^2 M_{00}^1 M_{00}^1 - 2st M_{00}^1 M_{00}^0 - t^2 M_{00}^0 M_{00}^0 \end{array} \right\}}} \\
&= \frac{sM_{00}^1 + tM_{00}^0}{\sqrt{s^2 M_{00}^2 M_{00}^0 - s^2 M_{00}^1 M_{00}^1}} \\
&= \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - M_{00}^1 M_{00}^1}} \frac{sM_{00}^1 + tM_{00}^0}{sM_{00}^1}
\end{aligned}$$

So we can conclude that $f_2(\Omega)$ and $f_3(\Omega)$ are invariant to both the photometric and geometric deformations. As a result, also their extrema will be reached in an invariant way.

Affinely invariant extrema of $f_I(t)$

For the intensity-based region extraction, we use the local extrema of the following function along rays emanating from a local intensity extremum:

$$f_I(t) = \frac{abs(|I(t) - I_0|)}{max(\frac{\int_0^t abs(|I(t) - I_0|)dt}{t}, d)}$$

with I_0 the intensity value at the extremum. It can easily be shown that this function is invariant to both geometric and photometric deformations. Let us start with the photometric ones:

$$\begin{aligned} f'_I(t) &= abs(|I'(t) - I'_0|) / \left(\frac{\int_0^t abs(|I'(t) - I'_0|) dt}{t} \right) \\ &= abs(|sI(t) + t - (sI_0 + t)|) / \left(\frac{\int_0^t abs(|sI(t) + t - (sI_0 + t)|) dt}{t} \right) \\ &= abs(|I(t) - I_0|) / \left(\frac{\int_0^t abs(|I(t) - I_0|) dt}{t} \right) \\ &= f_I(t) \end{aligned}$$

We only have to consider a scalefactor for the geometric deformations in this case, as we are looking along one-dimensional rays emanating from corresponding intensity extrema.

$$\begin{aligned} f'_I(t') &= abs(|I'(t') - I_0|) / \left(\frac{\int_0^t abs(|I'(t') - I_0|) dt'}{t'} \right) \\ &= abs(|I(t) - I_0|) / \left(\frac{\int_0^t abs(|I(t) - I_0|) s dt}{st} \right) \\ &= f_I(t) \end{aligned}$$

Hence, $f_I(t)$ is invariant to both the geometric and photometric deformations, and also its extrema are reached in an invariant way.

Appendix B

Derivation of a geometric semi-local constraint

Consider two images I and I'. Points in image I are denoted with homogeneous coordinates $\mathtt{p} = (x, y, z)^T$, while points in image I' are denoted with homogeneous coordinates $\mathtt{p}' = (x', y', z')^T$. A homography $\mathbf{H_i}$ belonging to a plane Π_i defines the following relation between the projections in images I and I' of 3D points lying on the plane Π_i

$$\mathtt{p}' = \mathbf{H_i}\mathtt{p}$$

with $\mathbf{H_i}$ a 3×3 matrix.

Consider two planes in 3D, Π_1 and Π_2, and their related homographies $\mathbf{H_1}$ and $\mathbf{H_2}$, and take an arbitrary point $\mathtt{p} = (x, y, z)^T$ in image I. Then, both $\mathbf{H_1}\mathtt{p}$ and $\mathbf{H_2}\mathtt{p}$ lie on the epipolar line corresponding to the point $\mathtt{p}$. Hence, the following formula for the epipolar line corresponding to the point $\mathtt{p}$ can be derived

$$l = (\mathbf{H_1}\mathtt{p}) \times (\mathbf{H_2}\mathtt{p})$$

where $\times$ denotes the vector product.
All epipolar lines pass through the same point $\mathtt{e}$, the epipole.

$$\exists \mathtt{e} \forall \mathtt{p} : (\mathbf{H_1}\mathtt{p} \times \mathbf{H_2}\mathtt{p})^T \mathtt{e} = 0$$

From this property, we can derive a constraint on $\mathbf{H_1}$ and $\mathbf{H_2}$.
If $\mathtt{H_{ij}}$ denotes the j-th column of matrix $\mathbf{H_i}$, this can be worked out as follows:

$$\exists \mathtt{e} \forall (x, y, z) : [(x\mathtt{H_{11}} + y\mathtt{H_{12}} + z\mathtt{H_{13}}) \times (x\mathtt{H_{21}} + y\mathtt{H_{22}} + z\mathtt{H_{23}})]^T \mathtt{e} = 0$$

This is a second-order equation in x, y and z with coefficients A, B, C, D, E and F functions of $\mathtt{e}$ and $\mathtt{H_{ij}}$.

$$\forall (x, y, z) : Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz = 0$$

Since this equation has to be fulfilled for all possible values x, y and z, all the coefficients in the equation have to be zero.

$$\begin{aligned} A &= (\mathtt{H}_{11} \times \mathtt{H}_{21})^T \mathtt{e} = 0 \\ B &= (\mathtt{H}_{12} \times \mathtt{H}_{22})^T \mathtt{e} = 0 \\ C &= (\mathtt{H}_{13} \times \mathtt{H}_{23})^T \mathtt{e} = 0 \\ D &= (\mathtt{H}_{11} \times \mathtt{H}_{22} + \mathtt{H}_{12} \times \mathtt{H}_{21})^T \mathtt{e} = 0 \\ E &= (\mathtt{H}_{11} \times \mathtt{H}_{23} + \mathtt{H}_{13} \times \mathtt{H}_{21})^T \mathtt{e} = 0 \\ F &= (\mathtt{H}_{12} \times \mathtt{H}_{23} + \mathtt{H}_{13} \times \mathtt{H}_{22})^T \mathtt{e} = 0 \end{aligned}$$

In order for all the above equations to have a solution $\mathtt{e} \neq (0,0,0)^T$, the following matrix, which is a function of $\mathtt{H}_{ij}$, must be rank-deficient.

$$rank \begin{pmatrix} (\mathtt{H}_{11} \times \mathtt{H}_{21})^T \\ (\mathtt{H}_{12} \times \mathtt{H}_{22})^T \\ (\mathtt{H}_{13} \times \mathtt{H}_{23})^T \\ (\mathtt{H}_{11} \times \mathtt{H}_{22} + \mathtt{H}_{12} \times \mathtt{H}_{21})^T \\ (\mathtt{H}_{11} \times \mathtt{H}_{23} + \mathtt{H}_{13} \times \mathtt{H}_{23})^T \\ (\mathtt{H}_{12} \times \mathtt{H}_{23} + \mathtt{H}_{13} \times \mathtt{H}_{22})^T \end{pmatrix} \leq 2$$

Applied to local regions

For local regions, the perspective deformation is too small to be detected. As a result, only an affine transformation can be derived. In this case, the homographies (from now on referred to as **A** and **B**) are of the following form:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

The rank-2 constraint derived in the previous section then becomes:

$$rank \begin{pmatrix} 0 & 0 & a_{11}b_{21} - b_{11}a_{21} \\ 0 & 0 & a_{12}b_{22} - b_{12}a_{22} \\ a_{23} - b_{23} & b_{13} - a_{13} & a_{13}b_{23} - b_{13}a_{23} \\ 0 & 0 & a_{11}b_{22} - b_{12}a_{21} + a_{12}b_{21} - b_{11}a_{22} \\ a_{22} - b_{22} & b_{12} - a_{12} & a_{12}b_{23} - b_{13}a_{22} + a_{13}b_{22} - b_{12}a_{23} \\ a_{21} - b_{21} & b_{11} - a_{11} & a_{11}b_{23} - b_{13}a_{21} + a_{13}b_{21} - b_{11}a_{23} \end{pmatrix} \leq 2$$

Rows (1), (2) and (4) force the epipole to lie at infinity. This corresponds to an orthographic projection model, which indeed leads to affine transformations between two views of a planar object. But also without forcing the epipole to infinity there is one constraint left:

$$rank \begin{pmatrix} a_{23} - b_{23} & b_{13} - a_{13} & a_{13}b_{23} - b_{13}a_{23} \\ a_{22} - b_{22} & b_{12} - a_{12} & a_{12}b_{23} - b_{13}a_{22} + a_{13}b_{22} - b_{12}a_{23} \\ a_{21} - b_{21} & b_{11} - a_{11} & a_{11}b_{23} - b_{13}a_{21} + a_{13}b_{21} - b_{11}a_{23} \end{pmatrix} \leq 2$$

The actual consistency constraint used in our experiments is then

$$det \begin{pmatrix} a_{23} - b_{23} & b_{13} - a_{13} & a_{13}b_{23} - b_{13}a_{23} \\ a_{22} - b_{22} & b_{12} - a_{12} & a_{12}b_{23} - b_{13}a_{22} + a_{13}b_{22} - b_{12}a_{23} \\ a_{21} - b_{21} & b_{11} - a_{11} & a_{11}b_{23} - b_{13}a_{21} + a_{13}b_{21} - b_{11}a_{23} \end{pmatrix} \leq \delta$$

with δ a predefined threshold.

Appendix C

Overview of the parameters

Parameters for anchor point detection

Harris corner detection parameters

N_{max}	estimated number of corners (upper limit)	300
N_{min}	estimated number of corners (lower limit)	200
σ	sigma of Gaussian smoothing kernel	0.7
	rel. minimum	$1e^{-5}$
k	scale factor	0.04
d	min. distance between two corner points	5
	adaptive window size	32
	density threshold	2

Local intensity extrema parameters

δ	window size for non-maximum suppression	10
σ	sigma of Gaussian smoothing kernel	3

Parameters for geometry-based region extraction

Canny edge detection parameters

sigma of Gaussian smoothing kernel	1
Gauss tail	0.015
lower hysteresis bound	6
upper hysteresis bound	12
min. edgel intensity	10
min. edge length	15
image border size	2

Line fitting parameters

min. line length	15
threshold	0.3
max. deviation in orientation	5

Region extraction

min. region size	5
max. region size	60
max. eccentricity	10

Parameters for intensity-based region extraction

number of rays	50
ray length	40

Parameters for matching of regions between two views

min. Mahalanobis-distance	2
min. crosscorrelation value (before finetuning)	0.7
min. crosscorrelation value (after finetuning)	0.8
max. number of iterations during finetuning	100
max. deviation in region parameters during finetuning	10
min. number of geometrically consistent matches	8
min. number of photometrically consistent matches	4

Appendix D

Overview of our database

Database retrieval results to a large degree depend on the images contained in the database and the images used to query the database. As we use a different definition of similarity than usual in the database retrieval literature, we could not use any of the standard databases, but had to build our own, which we could query with other images containing the same objects. As a result, before the reader can appreciate our results, we should first give him a more detailed overview of our database and the query images used.

Below, we give some image pairs, as used in our experiments. Each time, the query image is shown in the upper row, while the corresponding image in the database is shown in the lower row.




Duvel
Duvel
Coca-Cola
Coca-Cola


DE
PAPIEREN
TIJGER1

