Object Recognition Using Local Affine Frames

Štěpán Obdržálek

xobdrzal@fel.cvut.cz

September 30, 2006

Thesis Advisor: Doc. Dr. Ing. Jiří Matas

The research presented in this thesis was supported by Toyota Motor Corporation, the European Union project FP6-IST-004176 COSPAL, the Czech Ministry of Education projects MSM 210000012 and MSM 212300013, and by Czech Technical University grants CTU0209613 and CTU0307013.

Published by
Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
Object Recognition Using Local Affine Frames

A dissertation presented to the Faculty of the Electrical Engineering of the Czech Technical University in Prague in partial fulfilment of the requirements for the Ph.D. degree in Study Programme No. P2612 – Electrical Engineering and Information Technology, branch No. 3902V035 – Artificial Intelligence and Biocybernetics, by

Štěpán Obdržálek

September 29, 2006

Thesis Advisor

Doc. Dr. Ing. Jiří Matas

Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo Náměstí 13, 121 35 Prague 2, Czech Republic
Fax: +420 224 357 385, phone: +420 224 355 722
Abstract

Viewpoint-independent recognition of objects is a fundamental problem in computer vision. The common formulation is essentially: given some knowledge of how certain objects may appear, plus an image of a scene possibly containing those objects, find which objects are present in the scene and where. Recently, a considerable success in addressing the problem has been achieved by approaches based on matching of regions detected by processes that are locally quasi-invariant to viewpoint changes. This thesis proposes an object recognition method of this category – the goal of the thesis is to design a method which would recognise instances of rigid objects in large variety of scenes. The stress is put on recognition from largely different viewpoints and in different illumination conditions.

A complete object recognition system is presented, which includes content-driven extraction of repeatable image regions, extraction of local coordinate systems covariant with local affine transformations, geometrically and photometrically invariant representation of local object appearance, and an efficient organisation of the object database, which allows for fast recognition response. The system achieves close to real-time recognition and localisation of multiple objects in general scenes, and in performance compares well with other state-of-the-art methods.

Appearance of objects is represented by sets of measurements defined in local coordinate systems (local affine frames, LAFs) that are established on affine-covariantly detected image regions. A discrete cosine transform based descriptor of local appearance is proposed as a computationally and memory efficient representation, which in recognition performance is on par with the common SIFT representation. A new type of decision tree is proposed as a database organisation that supports matching in time sublinear with respect to the number of objects in the database, allowing therefore for scaling of the method to large datasets (thousands of objects).

State-of-the-art recognition results are presented on publicly available object recognition tests (COIL-100, ZuBuD, FOCUS). Changes of scale and illumination conditions, out-of-plane rotation, occlusion, local anisotropic scaling, and 3D translation of the viewpoint are all present in the test problems.
Resumé

Rozpoznávání objektů pozorovaných z různých pohledů je jedním ze základních otevřených problémů v počítačovém vidění. Formulace problému je následující: Víme-li, jak vypadají objekty jež nás zajímají a dostaneme-li obrázek scény ve které se mohou tyto objekty vyskytovat, cílem je určit které z objektů jsou ve scéně přítomny a kde. Výzkum v posledních letech ukázal, že úspěšného rozpoznávání dosahují metody, které reprezentují známé objekty sadou lokálních měření invariantních vůči lokálním změnám vzhledu při měnícím se pohledu. Objekt je rozpoznán, jsou-li nalezena korespondující měření mezi obrázky. Tato disertační práce navrhuje metodu rozpoznávání patřící právě do této skupiny. Cílem práce bylo navrhnout metodu, která bude rozpoznávat konkrétní objekty v co nejširším množství scén. Důraz byl kladen na umožnění značných změn pohledu na objekt a změn vzhledu objektu způsobených měnícími se okolními podmínkami, zvláště osvětlení.

Tato práce navrhuje kompletní systém pro rozpoznávání. Systém zahrnuje identifikaci takových oblastí v obrázcích, u nichž se očekává opakovaná detekce při změně pohledu, konstrukci lokálních souřadnic pro invariantních s afinitními změnami geometrie obrázku a geometricky a fotometricky invariantní reprezentaci lokálního vzhledu objektů. Takto získaná invariantní reprezentace objektů je uložena ve stromové struktuře umožňující rychlé vyhledávání. Navrhovaný systém rozpoznává a lokalizuje objekty v téměř reálném čase, s úspěšností dosahující či převyšující úspěšnost ostatních publikovaných přístupů.

Vzhled lokálních oblastí na objektu navrhovaly reprezentovat sadou nízkofrekvenčních koeficientů diskrétního diskrétního transformace. Tato reprezentace je pamatově úsporná a její výpočet je rychlý. V úspěšnosti rozpoznávání je srovnatelná s nyní de-facto standardní a nejběžnější použitou reprezentací ve formě SIFT, navržené Davidem Lowe. Pro rychlé vyhledávání v databázích objektů je použit nově navržený typ rozhodovacího stromu, který umožňuje vyhledání korespondencí mezi obrázky se sublineární časovou složitostí. Díky tomu je možné rozpoznávání objektů i z rozsáhlých datových sad (až tisíce objektů), aniž by docházelo k výraznému zpomalování systému.
Acknowledgement

I am indebted to my supervisor Jiří Matas for leading me throughout my research. His ideas, support, and considerable help have been of outmost importance for finishing my PhD study. He showed me how to conduct research and helped me with researching and writing of this thesis. My thanks go to my colleagues Michal Perdöch and Ondřej Chum, who I had the pleasure to collaborate with on closely related topics, who provided me with valuable inputs and ideas. I would also like to express my thanks to all my colleagues at Center for Machine Perception, especially to the head of the group Professor Václav Hlaváč, for creating an inspiring and a nice atmosphere.

The research presented in this thesis was supported by Toyota Motor Corporation, the European Union project FP6-IST-004176 COSPAL, the Czech Ministry of Education projects MSM 210000012 and MSM 212300013, and by Czech Technical University grants CTU0209613 and CTU0307013.
Contents

1 Introduction .................................................. 3
   1.1 Problem Formulation ....................................... 3
   1.2 Overview of the Proposed Approach ....................... 3
   1.3 Thesis Contributions ...................................... 4
   1.4 Structure of the Thesis .................................. 6

2 Overview and Related Work ................................... 7
   2.1 Appearance Based Approaches .............................. 7
   2.2 Geometry-Based Approaches ............................... 8
   2.3 Recognition via Correspondence of Local Features ........ 9
      2.3.1 The Approach of Lowe ............................... 11
      2.3.2 The Approach of Mikolajczyk & Schmid .............. 13
      2.3.3 The Approach of Tuytelaars & van Gool .......... 13
      2.3.4 The Approach of Zisserman et al. .................. 14
      2.3.5 The Approach of Selinger & Nelson ................. 15
      2.3.6 Other Related Work .................................. 16

3 Local Affine Frames ........................................ 21
   3.1 Geometric Invariance ..................................... 21
   3.2 Distinguished Regions .................................... 25
      3.2.1 Maximally Stable Extremal Regions (MSERs) .......... 25
      3.2.2 Ordering of Image Pixels ............................. 26
      3.2.3 Adaptation to Environmental Conditions ............ 28
   3.3 Geometric Primitives Covariant with Affine Transformations .. 31
      3.3.1 Proofs of Affine Covariance of LAF Constructions .. 34
      3.3.2 Details on Detection of the Geometric Primitives .. 36
   3.4 LAF Construction ......................................... 41
      3.4.1 Representation of Local Affine Frames ............... 43
   3.5 Reduction of the Number of Generated LAFs ................ 44
      3.5.1 Selection of LAFs Based on Uniform Image Coverage .... 44

4 Local Correspondences ...................................... 47
   4.1 Normalisation of Measurement Region .................... 47
   4.2 Descriptors of Local Appearance ......................... 49
   4.3 Matching: Forming Tentative Correspondences of Local Regions ... 51
   4.4 Globally Consistent Subset of Tentative Correspondences .... 53

5 Tentative Correspondences using Decision Trees ............... 55
   5.1 Introduction to the Decision Trees ...................... 55
   5.2 Recognition with Decision-Measurement Trees ........... 57
      5.2.1 Retrieval ........................................... 57
      5.2.2 Learning the Tree ................................... 58
   5.3 Concluding Remarks ...................................... 61
6 Experimental Validation 63
   6.1 Datasets ............................................. 63
      6.1.1 Mikolajczyk’s Dataset ............................. 63
      6.1.2 COIL-100 Dataset ................................. 64
      6.1.3 ZuBuD Dataset .................................... 64
      6.1.4 Focus Dataset ..................................... 65
      6.1.5 LAFsTest Dataset .................................. 65
   6.2 Evaluation of Components of the Recognition System .......... 65
      6.2.1 MSER Repeatability ............................... 66
      6.2.2 LAFs Repeatability ............................... 67
      6.2.3 Reduction of the Number of LAFs ..................... 71
      6.2.4 MSER Boundary Smoothing ......................... 75
      6.2.5 Required MSER Stability ........................... 76
      6.2.6 Measurement Region Size .......................... 77
      6.2.7 Discretisation of Measurement Region ............... 77
      6.2.8 Descriptors ...................................... 82
      6.2.9 Number of Low-Frequency DCT Coefficients .......... 86
   6.3 Performance Comparison with Published Results ............... 87
      6.3.1 Recognition Performance on COIL-100 Dataset .......... 87
      6.3.2 Recognition Performance on FOCUS Dataset ............ 87
      6.3.3 Recognition Performance on ZuBuD Dataset ........... 88
   6.4 Analysis of Computational Complexity ........................ 89

7 Object Recognition and Colour Constancy 91
   7.1 Interaction Between Object Recognition and Colour Constancy ...... 91
   7.2 Finding Global Photometric Transformation ..................... 92
   7.3 Experiments ........................................ 92
   7.4 Concluding Remarks .................................. 95

8 Conclusions 97

Bibliography 99
1 Introduction

1.1 Problem Formulation

Viewpoint-independent recognition of objects is a fundamental problem in computer vision. The common formulation is essentially: given some knowledge of how certain objects may appear, plus an image of a scene possibly containing those objects, find which objects are present in the scene and where. Consider the example in Figure 1.1. If the images in (a) represent objects of interest, which of the objects are present in scenes in (b)?

![Figure 1.1: Object recognition. (a) Examples of objects of interest, (b) examples of scenes where the objects might be sought.](image)

Recently, a considerable success in addressing the problem has been achieved by approaches based on matching of regions detected by processes that are locally quasi-invariant to viewpoint changes [Low04, MS02, MS01, TVG00, vZ03]. Such methods represent objects by sets of regions described by invariants computed from local measurements. The representation is learned from training images without manual intervention. During recognition, the same representation is built for the test image. The recognition problem is then formulated as a search for a geometrically consistent set of correspondences of regions from query and database images. Since it is not required that all local features match, the approaches are robust to occlusion and cluttered background. And since region-to-region correspondences are established, recognition also achieves localisation. This thesis proposes an object recognition method of this category.

The goal of the thesis was to design a method, which would recognise instances of rigid objects in large variety of scenes. The problem of categorisation was not considered. The stress was put on achieving recognition of objects observed from significantly different viewpoints and under different illumination conditions.

1.2 Overview of the Proposed Approach

The structure of the proposed recognition method is summarised in Algorithm 1 and visualised in Figure 1.2. Appearance of objects is represented by sets of measurements defined in local coordinate systems (Local Affine Frames, LAFs) that are established on affine-covariantly detected image regions. The LAFs are constructed by exploiting multiple affine-covariant procedures that take the detected regions as an input. Assuming locally planar approximation of object shape, any image measurement expressed in LAF coordinates is viewpoint-invariant. Appearance of the objects is thus represented by local patches, with shapes and locations given...
by the object-defined affine coordinate systems. The need for further transformation of image measurements to obtain invariant description, such as rotational or differential invariants, is eliminated.

**Algorithm 1** Structure of the MSER-LAF method

1. For every database and query image compute affine-covariant regions of data-dependent shape.
2. Construct local affine frames (LAFs) on the regions using multiple affine-covariant constructions.
3. Generate intensity representations of local image patches normalised according to the local affine frames. Photometrically normalise the patches.
4. Establish tentative correspondences between frames of query and database images. Compute similarity between the patches, select most similar pairs.
5. Find a globally consistent subset of the correspondences. Infer the presence and location of the objects.

1.3 Thesis Contributions

The following list summarises the main contributions of the thesis.

- **Complete LAF-based object recognition framework.** A complete object recognition system is presented, which includes extraction of repeatable image regions, extraction of local coordinate systems covariant with local affine transformations, geometrically and photometrically invariant representation of local image appearance, and an efficient organisation of the object database, which allows for fast recognition response. The system achieves close to real-time recognition and localisation of multiple objects in general scenes, and in performance compares well with other state-of-the-art methods.

- **Local Affine Frames.** Different affine-covariant geometric primitives are categorised, their covariance is theoretically proven, and computational details are given. Combining these primitives, local coordinate systems are constructed and used to extract affine-invariant measurements from images. (Section 3.3).

- **A new type of decision tree for object representation.** A database organisation is proposed that supports matching in time sublinear with respect to the number of objects in the database, therefore allows for scaling of the method to large databases (Chapter 5). The tree is optimised for minimal retrieval time, and localisation uncertainty of LAF detection is explicitly considered.

- **A discrete cosine transform based descriptor of local appearance.** The DCT representation is computationally and memory efficient (Section 4.2). The recognition performance is on par with the standard SIFT representation.

- **Multiple orderings of colour pixels.** Not only the MSER region detector used in this work, but almost any other detector, e.g. of Harris interest points, relies on pixels being from
1.3 Thesis Contributions

an ordered set. This is traditionally achieved by ordering colour pixels by intensity (e.g. by converting images to greyscale), but different orderings can be used (Section 3.2.2).

- Runtime adaptation of the recognition process. Characteristics of actual runtime environmental conditions are estimated from statistics gathered on successful object recognitions, and subsequent recognition is adapted to reflect them (Section 3.2.3).

- Reduction of highly redundant image representation. The established LAFs are ordered by their added value to object representation, and least valuable frames are discarded. The process maintains a representative coverage of the image, i.e. frames are first removed from over-represented locations (Section 3.5).
• Solution to colour constancy problem in scenes where successful object recognition is achieved (Chapter 7).

• Recognition performance is demonstrated on multiple challenging recognition problems, and experimental evaluation of individual components of the system is presented (Chapter 6).

1.4 Structure of the Thesis

The thesis is structured as follows. Chapter 2 gives an overview of the state of the art in object recognition. Relevant published approaches are categorised and summarised. In Chapter 3, an overview and a taxonomy of affine-covariant constructions of local coordinate systems (frames) are presented, the affine covariance of the constructions is proven, and computation issues discussed. Chapter 4 describes the process of geometric and photometric normalisation of local appearance and the forming of local region-to-region correspondences. Chapter 5 proposes a computationally efficient solution to the correspondence formation task. Sublinear retrieval time, necessary for a large-scale recognition problems, is achieved using decision trees. In Chapter 6, several implementation decisions about components of the system are experimentally evaluated and their impact on the recognition rate is shown. State-of-the-art results are presented on publicly available object recognition tests (COIL-100, ZuBuD, FOCUS). Changes of scale and illumination conditions, out-of-plane rotation, occlusion, local anisotropic scaling, and 3D translation of the viewpoint are all present in the test problems. Finally, Chapter 7 discusses the relation between colour constancy and object recognition, and how successful object recognition can be exploited to solve the colour constancy problem.
2 Overview and Related Work

Recognition of three-dimensional objects in 2D images and videos is a challenging problem. Given an image of an unknown scene, the task is to identify and localise known objects which the scene might contain. The recognition is accomplished by matching features of the image and a model of an object. The two most important issues that a method must address are the definition of a feature, and how the matching is found.

What do we consider the most desirable properties of a recognition system? Generality, i.e. the ability to recognise any object without hand-crafted adaptation to a specific task, robustness, the ability to recognise the objects in variable conditions, and simple learning, i.e. avoiding special or demanding procedures to obtain the database of models. Obviously these requirements are generally impossible to achieve, as it is for example impossible to recognise objects in images taken in complete darkness. The challenge is then to develop a method with minimal constraints.

Object recognition methods can be classified according to a number of characteristics. We focus on model acquisition (learning) and invariance to image formation conditions. Historically, two main groups can be identified. In the so called geometry- or model-based object recognition, the knowledge of an object appearance is provided by the user as an explicit CAD-like model. Typically, such a model describes only the 3D shape, omitting other properties such as colour and texture. On the other end of the spectrum are pure appearance-based methods, where no explicit user-provided model is required. The object representations are usually acquired through an automatic learning phase (but not necessarily), and the model typically relies on surface reflectance (albedo) properties. Recently, methods which put into correspondence local image patches have emerged. Models are learned automatically, objects are represented by appearance of small local elements, and the global arrangement of the representation is constrained by weaker or stronger geometric models.

2.1 Appearance Based Approaches

The central idea behind appearance-based methods is the following. Having seen all possible appearances of an object, can recognition be achieved by just efficiently remembering all of them? Could recognition be thus implemented as an efficient visual (pictorial) memory? The answer obviously depends on what is meant by "all appearances". The approach has been successfully demonstrated for scenes with unoccluded objects on black background, e.g. [NWN96]. But remembering all possible object appearances in the case of arbitrary background, occlusion and illumination, is currently computationally prohibitive for collections with non-trivial number of objects.

Appearance based methods [BBN+98, TP91, LB96, BHK97, PMS94, NdS02, SW96, LB00, MP95, NWN96] typically include two phases. In the first phase, a model is constructed from a set of reference images. The set includes the appearance of the object under different orientations, different illuminants and potentially multiple instances of a class of objects, for example human faces. The images are highly correlated and can be efficiently compressed using e.g. Karhunen-Loeve transformation (also known as Principal Component Analysis - PCA).

In the second phase, 'recall', parts of the input image (subimages of the same size as the training images) are extracted, possibly by segmentation (by texture, colour, motion) or by exhaustive enumeration of image windows over whole image. The recognition system then compares an extracted part of the input image with the reference images (e.g. by projecting the
part to the Karhunen-Loeve space).

A major limitation of the appearance-based approaches is that they require isolation of the complete object of interest from the background. They are thus sensitive to occlusion and require good segmentation. A number of attempts have been made to address recognition with occluded or partial data [MN95, MP95, SL03, BWL01, LB00, BL98, SBL02, LB96, JL00, Kru96].

The family of appearance-based object recognition methods includes approaches based on matching of global image characteristics. In [SB90, SB91], Swain and Ballard proposed to represent an object by a colour histogram. Objects are identified by matching histograms of image regions to histograms of a model image. While the technique is robust to object orientation, scaling, and occlusion, it is very sensitive to lighting conditions, and it is not suitable for recognition of objects that cannot be identified by colour alone. The approach has been later modified by Healey and Slater [HS94] and Funt and Finlayson [FF95] to exploit illumination invariants. Recently, the concept of histogram matching was generalised by Schiele [SC00, SC96b, SC96a], where, instead of pixel colours, responses of various filters are used to form the histograms (called then receptive field histograms). Later it evolved into the “bag of features” methodology (histograms of local characteristics), currently popular in categorisation.

To summarise, appearance-based approaches are attractive since they do not require image features or geometric primitives to be detected and matched. But their limitations, i.e. the necessity of dense sampling of training views and the low robustness to occlusion and cluttered background, make them suitable mainly for certain applications with limited or controlled variations in the image formation conditions, e.g. for industrial inspection.

2.2 Geometry-Based Approaches

In geometry- (or shape-, or model-) based methods, the information about the objects is represented explicitly. The recognition can than be interpreted as deciding whether (a part of) a given image can be a projection of the known (usually 3D) model [Pop94] of an object.

Generally, two representations are needed: one to represent object model, and another to represent the image content. To facilitate finding a match between model and image, the two representations should be closely related, in the ideal case there will be a simple relation between primitives used to describe the model and those used to describe the image. Would the object be, for example, described by a wireframe model, the image might be best described in terms of linear intensity edges. Each edge can be then matched directly to one of the model wires. However, the model and image representations often have distinctly different meanings. The model may describe the 3D shape of an object while the image edges correspond only to visible manifestations of that shape, mixed together with ’false’ edges (discontinuities in surface albedo) and illumination effects (shadows). To achieve pose and illumination invariance, it is preferable to employ model primitives that are at least somewhat invariant with respect to changes in these conditions. Considerable effort has been directed to identify primitives that are invariant with respect to viewpoint change [MZ92, Wei93].

Model primitives are organised typically into some sort of structure. A frequent approach is to arrange them hierarchically according to part/whole relations. Levels of the hierarchy represent degrees of grouping and primitives occur only at the hierarchy’s lowest level [Ett88]. Another method of organizing shape primitives is to arrange them according to adjacency relations so that each primitive is related to others nearby [Sha80, WLR89]. The structure is often represented as a graph in which nodes denote primitives or groups of primitives, and edges denote the relations among them. Recognition then becomes the problem of graph matching, where a structure retrieved from an image is matched against known, user-modelled structures.

The main disadvantages of geometry-based methods are the dependence on reliable extraction
of geometric primitives (lines, circles, etc.), the ambiguity in interpretation of the detected primitives (presence of primitives that are not modelled), the restricted modelling capabilities only to a class of objects which are composed of few easily detectable elements, and the need to create the models manually.

2.3 Recognition via Correspondence of Local Features

Neither geometry-based nor appearance-based methods discussed previously perform well regarding the requirements stated at the beginning of the chapter, i.e. the *generality*, *robustness*, and *simple learning*. Geometry-based approaches require the user to specify the object models, and can usually handle only objects consisting of simple geometric primitives. They are not general, nor do they support simple learning. Pure appearance-based methods demand exhaustive set of learning images, taken from densely sampled views and under a variety of illuminations. Such set is only available when the object can be observed in a controlled environment, e.g. placed on a turntable. The methods are also sensitive to occlusion of the objects and to the unknown background, thus they are not robust.

As an attempt to address the above mentioned issues, methods based on matching of local features have been proposed. Objects are represented by a set of local features, which are automatically computed from training images. The learned features are organised into a database. When recognising a query image, local features are extracted as in the training images. Similar features are then retrieved from the database and the presence of objects is assessed in the terms of the number of local correspondences. Since it is not required that all local features match, the approaches are robust to occlusion and cluttered background.

To recognise objects from different views, it is necessary to handle global variations in object appearance. The variations might be complex in general, but at the scale of the local features they can be modelled by simple, e.g. affine, transformations. Thus, by allowing simple transformations at local scale, a significant viewpoint invariance is achieved even for objects with complicated shapes. As a result, it is possible to obtain models of objects from only a few views, e.g. taken 90 degrees apart.

The main advantages of approaches based on matching local features are summarised below.

- Learning, i.e. the construction of internal models of known objects, is done automatically from images depicting the objects. No user intervention is required except for providing the training images.
- The local representation is based on appearance. There is no need to extract geometric primitives (e.g. lines), which are generally hard to detect reliably.
- Segmentation of objects from background is not required prior to recognition, and yet objects are recognised on an unknown background.
- Objects of interest are recognised even if partially occluded by other unknown objects in the scene.
- Complex variations in object appearance caused by varying viewpoint and illumination conditions are approximated by simple transformations at a local scale.
- Measurements on both database and query images are obtained and represented in an identical way.

Putting local features into correspondence is an approach that is robust to object occlusion and cluttered background in principle. When other objects in the scene occlude a part of an
object, only features of that part are missed. As long as there are enough features detected in the unoccluded part, the object can be recognised. The problem of cluttered background is solved in a final step of the recognition process, when a hypothesised match is verified and confirmed on the basis of global geometric consistency, and false correspondences are rejected.

Several approaches based on local features have been proposed. Generally, they follow a certain common structure, components of which are summarised below.

Detectors. First, image elements of ‘interest’ are detected. The elements will serve as anchor locations in the images – descriptors of local appearance will be computed at these locations. Thus, an image element is of interest if it depicts a part of an object, which can be repeatedly detected and localised in images taken over large range of conditions. The challenge is to find such a definition of the ‘interest’, that would allow fast, reliable and precisely localised detection of such elements. The brute force alternative to the detectors is to generate local descriptors at every point.

Descriptors. Once the elements of interest are found, the image appearance in their neighbourhood has to be encoded in a way that would allow for searching of similarly appearing elements.

When designing a descriptor (sometimes called a feature vector), several aspects have to be taken into account. First, the descriptors should be discriminative enough to distinguish between features of the objects stored in the database. Would we for example want to distinguish between two or three objects, each described by some ten-odd local features, the descriptions of local appearance can be as simple as e.g. four-bin colour histograms. On the other hand, handling thousands of database objects requires the ability to distinguish between a vast number of descriptors, demanding thus highly discriminative representation. This problem can be partially alleviated by using grouping, i.e. simultaneous consistent matching of several detected elements.

Another aspect in designing a descriptor is that it has to be invariant, or at least in some degree robust, to geometric variations that are not reflected by the detector. If, for example, the detector detects circular or elliptical regions without assigning an orientation to them, the descriptor must be made invariant to the orientation (rotational invariants). Or, if the detector is imprecise in locating the elements of interest, e.g. having few pixel tolerance, the descriptor must be insensitive to these small misalignments. Such a descriptor might be based e.g. on colour moments (integral statistics over whole region), or on local histograms.

It follows that the major factors of a recognition system that affect the discriminative potential, and thus the ability to handle large object databases, are the repeatability and the localisation precision of the detector.

Indexing. During learning of object models, descriptors of local appearance are stored into a database. In the recognition phase, descriptors are computed on the query image, and the database is looked up for similar descriptors (potential matches). The database should be organised (indexed) in a way that allows an efficient retrieval of similar descriptors. The character of suitable indexing structure depends generally on the properties of the descriptors (e.g. their dimensionality) and on the distance measure used to determine which are the similar ones (e.g. euclidean distance). For optimal performance of the index (fast retrieval times), such combination of descriptor and distance measure should be sought, that maximises the ratio of similarities between correctly and of falsely matched descriptors [Low04].
The choice of indexing scheme has major effect on the speed of the recognition process, especially on how the system scales to large object databases. Commonly, though, the database searches are done simply by sequential scan, i.e. without using any indexing structure.

**Matching.** When recognising objects in an unknown query image, local features are computed in the same form as for the database images. None, one, or possibly more tentative correspondences are then established for every feature detected in the query image. Searching the database, euclidean or mahalanobis distance is typically evaluated between the query feature and the features stored in the database. The closest match, if close enough, is retrieved. These tentative correspondences are based purely on the similarity of the descriptors. A database object which exhibit high (non-random) number of established correspondences is considered as a candidate match.

**Verification.** The similarity of descriptors, on its own, is not a measure reliable enough to guarantee that an established correspondence is correct. As a final step of the recognition process, a verification of presence of the model in the query image is performed. A global transformation relating the images is estimated in a robust way (e.g. by using the RANSAC algorithm). Typically, the global transformation has the form of epipolar geometry constraint for general (but rigid) 3D objects, or of homography for planar objects. More complex transformations can be derived for non-rigid or articulated (piecewise rigid) objects.

As a result, tentative correspondences which are not consistent with the estimated global transformation are rejected, and only remaining correspondences are used to estimate the final score of the match.

As mentioned before, if a detector cannot recover certain parameters of the image transformations, descriptor must be made invariant to them. It is preferable, though, to have a covariant detector rather than an invariant descriptor, as that allows for more powerful global consistency verification. If, for example, the detector does not provide the orientations of the image elements, rotational invariants have to be employed in the descriptor. In such a case, it is impossible to verify that all of the matched elements agree in their orientation.

In the following, main contributions to the field of object recognition based on local correspondences are reviewed. The approaches follow the aforementioned structure, but differ in individual steps; in the way how are the local features obtained (detectors), and what are the features themselves (descriptors).

### 2.3.1 The Approach of Lowe

David Lowe has developed an object recognition system [BL99, Low99, BL03, BL97, BL02, Low01, Low04], with emphasis on efficiency, achieving real-time recognition times. Anchor points of interest are detected with invariance to scale, rotation and translation. Since local patches undergo more complicated transformations than similarities, a local-histogram based descriptor is proposed, which is robust to imprecisions in alignment of the patches.

**Detector.** The detection of regions of interest proceeds as follows:

1. Detection of scale-space extrema. Circular regions, which have maximal response of the Difference-of-Gaussians (DoG) filter, are detected at all scales and image locations. Efficient implementation exploits scale-space pyramid.
The scale space of an image $I$ is defined as a function, $L(x,y,\sigma)$, that is produced from the convolution of a variable-scale Gaussian, $G(x,y,\sigma)$, with an input image, $I(x,y)$:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$ (2.1)

where $*$ is the convolution operation in $x$ and $y$, and

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$ (2.2)

The difference-of-gaussians scale space is then defined as

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma)$$ (2.3)

where $k$ is a constant factor.

The initial image is repeatedly convolved with Gaussian filter to produce a set of scale-space images. Adjacent scale-space images are then subtracted to produce a set of difference-of-gaussians images. In these images, local minima and maxima (i.e. extrema of the DoG filter response) are detected, both in spatial an scale domains. The result of the first phase is thus a set of triplets $x, y$ and $\sigma$, image locations and a characteristic scales.

2. The location of the detected points is refined. The DoG responses are locally fitted with 3D quadratic function and the location and characteristic scale of the circular regions are determined with subpixel accuracy. The refinement is necessary, as, at higher levels of the pyramid, a displacement by a single pixel might result in a large shift in the original image domain. Unstable regions are then rejected, the stability is given by the magnitude of the DoG response. Regions with the response lower than a predefined threshold are discarder. Further regions are discarded that were found along linear edges, which, although having high DoG response, have unstable localisation in one direction.

3. One or more orientations are assigned to each region. Local histograms of gradient orientations are formed and significant peaks in the histogram determine the characteristic orientations.

The SIFT Descriptor. The DoG regions are detected invariantly to scaling and rotation, i.e. to transformations from the similarity group. The description of the regions accommodates the changes in appearance that are not well modelled by similarities. Local image gradients are measured at the region’s characteristic scale, weighted by the distance from the region’s centre and combined into a set of orientation histograms. Using the histograms, small misalignments in the localisation does not affect the final description. The construction of the descriptors allows for approximately $20^\circ$ 3D rotations before the similarity model fails. At the end, every detected region is represented by a 128-dimensional vector.

Indexing. To support fast retrieval of database vectors, a modification of the $k$D tree algorithm, called BBF (best bin first), is adopted. The algorithm is approximate in the sense that it returns the closest neighbour with high probability, or else another point that is very close in distance to the closest neighbour. The BBF algorithm modifies the $k$D tree algorithm to search bins in the order of their closest distance from the query location, instead of the order given by the tree hierarchy.

Verification. The Hough transform is used to identify clusters of tentative correspondences with a consistent geometric transformation. Since the actual transformation is approximated by a similarity, the Hough accumulator is 4-dimensionional and is partitioned to rather broad bins. Only clusters with at least 3 entries in a bin, are considered further. Each such cluster is then subject to a geometric verification procedure in which an iterative least-squares fitting is used to find the best affine projection relating the query and database images.
2.3 Recognition via Correspondence of Local Features

2.3.2 The Approach of Mikolajczyk & Schmid

The approach of Schmid et al. is described in [RLSP03, MS02, SM97, SM96a, Sch01, SM96b, MS01, DS03]. Based on an affine generalisation of Harris corner detector, anchor points are detected, and shape-adapted elliptical neighbourhoods are described by Gaussian derivatives of image intensities.

**Detector.** Mikolajczyk and Schmid built on the work of Lindeberg [Lin98] and Baumberg [Bau00] and implemented an affine-adapted Harris point detector. The affine scale space of an image $I$ is defined similarly as in eq. 2.1: $L(x, y, \Sigma)$ is produced by convolving a variable-shaped Gaussian, $G(x, y, \Sigma)$, with an input image, $I(x, y)$:

$$L(x, y, \Sigma) = G(x, y, \Sigma) * I(x, y)$$  \hspace{1cm} (2.4)

where $*$ is the convolution operation in $x$ and $y$, and

$$G(x, y, \Sigma) = \frac{1}{2\pi\sqrt{\det\Sigma}} e^{-\frac{(x,y)^T\Sigma^{-1}(x,y)^T}{2}}.$$  \hspace{1cm} (2.5)

The parameter $\Sigma$ is a positive-definite $2 \times 2$ matrix with three free parameters. Together with location $x, y$, the parametric space is five-dimensional, which is too complex to be computationally feasible. They therefore propose a solution which starts with points detected in uniform scale space and iteratively search for affine shape adaptation of their neighbourhoods.

For initialisation, approximate locations and scales of interest points are extracted by standard multi-scale Harris detector. These points are not affine invariant because of the uniform Gaussian kernel used. Given the initial location solution, their algorithm iteratively modifies the shape, the scale and the spatial location of neighbourhood of each point, and converges to affine-invariant interest points. For more details see [MS02].

**Descriptors and Matching.** The descriptors are composed from Gaussian derivatives computed over the shape-adapted elliptical regions. Invariance to rotation is obtained by ‘steering’ the derivatives in the direction of gradient, which is estimated as the average gradient orientation in the point’s neighbourhood. Using derivatives up to 4th order, the descriptors are 12-dimensional, and the similarity of descriptors is evaluated in Mahalanobis distance. Promising matches are then confirmed or rejected by cross-correlation computed over normalised neighbourhood windows. Recently they adopted the SIFT descriptor of David Lowe.

**Verification.** Once the point-to-point correspondences are obtained, a robust estimation of geometric transformation is computed using RANSAC algorithm. The transformation class used is either a homography or a fundamental matrix.

Recently, Dorko and Schmid [DS03] extended the approach towards object categorisation. Local image patches are detected and described by the same approach as described above. Patches from several examples of objects from a given category (e.g. cars) are collected together, and a classifier is trained to distinguish them from patches of different categories and from background patches.

2.3.3 The Approach of Tuytelaars & van Gool

Luc van Gool and his collaborators developed another approach based on matching of local image features [TVG00, TG99, FTVG03, TTVG03, TGDK99, Tuy00, TTMVG01]. They start with detection of elliptical or parallelogram-shaped image regions. The regions are described by a vector of photometrically invariant generalised colour moments, and matching is typically verified by the epipolar geometry constraint.

**Detector.** Two methods for extraction of affinely invariant regions are proposed, yielding so-called geometry- and intensity-based regions. The regions are affine covariant, they adapt
their shape to the underlying image intensity in order to keep on representing the same physical part of an object. Apart from the geometric invariance, photometric invariance allows for independent scaling and offsets for each of the three colour channels.

The region extraction always starts by detecting stable anchor points. The anchor points are either Harris points [HS88], or local extrema of image intensity. Although the detection of Harris points is not really affine invariant, as the support set over which the response is computed is circular, the points are still fairly stable under viewpoint changes. Intensity extrema, on the other hand, are invariant to any continuous geometric transformation and to any monotonic transformation of the intensity, but they are not localised as accurately.

**Geometry-based regions** Starting from a detected Harris point, edges obtained by Canny detector in the point's neighbourhood are exploited. Parallelograms are formed which have one corner at the anchor point, and two other corners on two different edges. The position of the two corners is not arbitrary, their relative distances along the edges are constrained by an affine invariant term. Such parallelograms are then chosen, where a specific function of the underlying image intensities is strongly extremal (see [Tuy00] for details). The geometry-based regions are covariant with affine transformations.

**Intensity-based regions** To avoid the inherent unreliability of the edge detections, pure intensity-based region extraction is also proposed. Pencil of rays is emanated from each intensity extrema, and intensity profile is studied along each of the rays. On each ray a point is sought which correspond to abrupt changes in the intensity, i.e. to locations where the rays leave a well delimited region of stable intensities. The points found on each of the rays are then connected to an irregularly shaped region. The region is later approximated by an elliptical region preserving shape moments up to the second order. The intensity-based regions do not resolve affine transformations fully. One degree of freedom, the rotation within the ellipse, is not determined.

On colour images, the detection is performed thrice, separately on each of the colour bands.

**Descriptors and Matching.** In the case of geometry-based regions, each of the regions is described by a vector of 18 generalised colour moments [MMG99], invariant to photometric transformations. For the intensity-based regions, 9 rotation-invariant generalised colour moments are used. The similarity between the descriptors is given by the Mahalanobis distance, correspondences between two images are formed from regions with the distance mutually smallest. Once corresponding regions have been found, the cross-correlation between the underlying image intensities (not the moments) is computed as a final check before accepting the match. In the case of the intensity-based regions, where the rotation is unknown, the cross-correlation is maximised over all rotations. Good matches are further fine-tuned by non-linear optimisation: the cross-correlation is maximised over small deviations of the transformation parameters.

**Verification.** The set of tentative correspondences is pruned by both geometric and photometric constraints. The geometric constraint basically rejects correspondences contradicting the epipolar geometry. Photometric constraint assumes that there is always a group of corresponding regions that undergo the same transformation of intensities. Correspondences that have singular photometric transformation are rejected. Recently an extension to the verification step was published [FTG04], which allows for precise model localisation and segmentation even for flexible, nonrigid objects.

### 2.3.4 The Approach of Zisserman et al.

A. Zisserman and his collaborators developed strategies for matching of local features mainly in the context of the wide-baseline stereo problem [PZ98b, PZ98a, SZ02b, SZ98, SZ01]. Recently
they presented an interesting work relating the image retrieval problem to text retrieval [vZ03, SZ02a, SZ03]. They introduced an image retrieval system, called VideoGoogle, which is capable of processing and indexing full-length movies.

**Detectors and Descriptors.** Two types of detectors of anchor elements are employed. One is the shape-adapted elliptical regions by Mikolajczyk and Schmid, as described in Section 2.3.2, second the Maximally Stable Extremal Regions described later in Section 3.2.1. Local representation is realised by the SIFT descriptors introduced by David Lowe (see Section 2.3.1). Knowing that a motion video sequence is being processed, noisy and unstable regions can be eliminated by temporal filtering. The regions detected in each frame of the video are tracked using a simple constant velocity dynamic model and correlation. Any region, which does not survive for more than three consecutive frames is rejected. Descriptor of a region is averaged over the track.

**Indexing and Matching.** The descriptors are grouped into clusters, based on their similarity. In analogy to stop-lists in text retrieval, where common words, like ‘the’, are ignored, large clusters are eliminated. When a new image is observed, each descriptor of the new image is matched only against representants of individual clusters. Selection of the nearest cluster immediately generates matches for all frames of the cluster, throughout the whole movie. The exhaustive comparison with every descriptor of every frame is thus avoided. The similarity measure, used for both the clustering and the closest cluster determination, is the Mahalanobis distance of SIFT descriptors.

**Verification.** Video frames are first retrieved using the frequency of matched descriptors, and then re-ranked based on spatial consistency of the correspondences. A search area of each match is defined by few nearest neighbours. Other regions, which also match within this area casts a vote for that match. Matches with no support are rejected, the final rank of the frame is determined by the total number of votes.

**2.3.5 The Approach of Selinger & Nelson**

The object recognition system developed by Randal Nelson and Andrea Selinger at the University of Rochester exploits a four-level hierarchy of grouping processes [NS98, SN01a, Sel01, SN00, SN99, SN01b]. The system architecture is similar to other approaches recapitulated here, though a different terminology is used. Inspired by the Gestalt laws and perceptual grouping principles, a four-level grouping hierarchy is built, where higher levels contains groups of elements from lower levels.

The hierarchy is constructed as follows. At the fourth (highest) level, a 3D object is represented as a topologically structured set of flexible 2D views. The geometric relation between the views is stored here. This level is used for geometric reasoning, but not for recognition. Recognition takes place at the third level, the level of individual views. In these views the visual appearance of an object, derived from a training image, is represented as a loosely structured combination of a number of so-called local context regions. Local context regions (local features) are represented at the second level. The regions can be thought of as local image patches that surround first level features. At the first level are features (detected image elements) that are around ”strong” first level features are context patches (the second level) constructed.

Efficient recognition is achieved by using a database implemented as an associative memory of context patches. An unknown context patch recalls associated hypotheses for all known views of objects that could have produced such a context patch. These hypotheses are processed by a second associative memory, indexed by view parameters, which partitions the hypotheses into clusters that are mutually consistent within a loose geometric framework (these clusters are the
third level groups). The looseness is obtained by tolerating a specified deviation in position, size, and orientation. The bounds are set to be consistent with a given distance between training views (e.g. approximately 20 degrees). The output of the recognition stage is a set of third level groupings that represent hypotheses of the identity and pose of objects in the scene, ranked by total evidence for each hypothesis.

2.3.6 Other Related Work

Scale Saliency by Kadir & Brady

Kadir and Brady presented a generalisation of the concept of detector of image elements of interest [KB01]. The model underlying their algorithm deems image regions salient if they are unpredictable in given descriptor-space, i.e. if exhibiting high entropy with respect to a chosen representation of local appearance. The approach offers a more general model of feature saliency compared with conventional techniques, which define saliency only with respect to a particular set of properties, chosen in advance.

In its basic form, the algorithm is invariant only to similarity transformations (thence the name 'scale' saliency; only the scale of circular regions is estimated on top of their locations). The approach is invariant to intensity shifts, and is robust to noise, small changes in viewpoint, and intensity scalings. They formulate the Scale Saliency a product of two terms, each a function of the PDF of local image descriptor at multiple scales. First term is the Shannon entropy of the descriptor at a particular image location in a given descriptor-space. The second term, called the inter-scale saliency measure, represents a measure of the magnitude change of the local PDF as a function of scale. Regions are detected at locations and scales where both of the terms peak.

Recently, an affine extension to the scale selection was presented [KB03], capable of detecting elliptical regions. The modified saliency measure is then a function of three parameters representing the affine deformation, instead of the single one for the scale. It is noted however, that the straightforward modification of the algorithm is rather sensitive to image noise, and alternative formulations of the affine saliency are being sought.

Local PCA, approaches of Jugessur and Ohba

As discussed in Section 2.1, global PCA (principal component analysis) based methods are sensitive to variations in the background behind objects of interest, changes in the orientation of the objects, and to occlusion. Traditional global approaches fail to recognize objects successfully if more than about one third of image differs. Ohba and Ikeuchi [OI97] and Jugessur and Dudek [JD00] propose an appearance-based object recognition method robust to variations in the background and occlusion of a substantial fraction of the image. In order to apply the eigenspace analysis to recognition of partially occluded objects, they propose to divide the object appearance into small windows, referred to as 'eigen windows' [OI97], and to apply eigenspace analysis to them. Like in other approaches exploiting local appearance, even if some of the windows are occluded, the remaining are still effective and can recover the object identity and pose.

An important issue in the eigen window technique is the selection of the optimal set of the windows. Ohba and Ikeuchi [OI97] introduce three criteria to select the windows: detectability, uniqueness, and reliability. The detectability measures how easy it is to detect and localise the window within a large image, e.g. that a window containing corners of an object is much easier to localise than those containing a planar textureless region. To select only discriminative windows, a measure of uniqueness is evaluated, and detected windows that are similar to others
are rejected, e.g. in the case that target object has multiple similar components. Finally, a reliability measure selects windows that remain stable within a range of object poses.

In similar approach, Jugessur and Dudek [JD00] employ an interest operator which chooses points within the images. Sub-windows are then created around the chosen points and PCA is computed on them. Only discriminative windows are held, the discriminativity of a window is determined by computing the standard deviation of intensity values within that window. In addition to robustness to occlusions, they also address the problem of rotation invariance. The proposed solution is to compute the PCA not on the intensity patches, but rather in frequency domain of windows represented in polar coordinates.

**Maximally Stable Corner Points by Fraundorfer**

Fraundorfer et al. [FSB06, FWB05] use a correspondence search approach for robot navigation. A piece-wise planar world map is built, composed of landmarks in form of small planar patches localised in 3D. Robot location is then implied from spatial configuration of the map patches that were matched against current visual input. Each patch is associated with a SIFT-descriptor and with the original appearance from the image. As the anchor detector, a novel local detector, Maximally Stable Corner Cluster (MSCC) detector, is proposed. MSCC regions are formed by multi-scale clusters of Harris corner points. Each detected cluster represents a distinguished region, cluster centres define the positions of the detection, outline of the regions is defined by cluster borders. Comparison with other detectors has shown that the MSCC detector largely detects regions at different image locations, thus it allows for effective combinations with other state-of-the-art methods.

**Hyper-Polyhedron Indexing of Shao & Svoboda**

Hao Shao and Tomáš Svoboda [SSF03, SSTG03a] extended the work of Tuytelaars and Luc van Gool (see Section 2.3.3), introducing an efficient indexing scheme for faster image retrieval from large databases. As in the Tuytelaar’s approach, images are processed by detector of intensity-based regions, which are covariant with affine transformations up to an unknown rotation. Nine rotation-invariant colour moments are computed for each region. The database images are thus represented by a set of points in a 9-dimensional space. During the recognition phase, identical 9-dimensional descriptors are computed for a query image, and matching is performed by finding database points closest in Mahalanobis distance.

To simplify the indexing structure, the Mahalanobis distance is avoided by first transforming the space of database descriptors according to the covariance matrix, so that the Mahalanobis distance becomes Euclidean. A neighbour search algorithm is proposed that is inspired by the approach by Nene and Nayar [NN97]. There a hyper-cube is used as an approximation of the hyper-sphere surrounding a query point (or, points are retrieved which are close in L1 metric instead of L2). The algorithm is very simple and begins with selecting the points that are in first dimension closer than a radius \( r \) to the query point. Selected points are added to a ‘candidate’ list. Next, it trims the candidate list by discarding points that are farther than \( r \) in the second dimension. This procedure is repeated for each dimension to end up with a list of points in a hypercube of size \( 2r \), centred on the query point. The closest point is then found by exhaustive search within the list. The hyper-cube approximation of the hyper-sphere deteriorates very quickly with increasing dimensions, in the case of 9 dimensions, the volume of the hyper-sphere is only 0.7% of the hyper-cube volume, which results in unnecessary computation load during the exhaustive search phase.

Shao and Svoboda propose a more accurate approximation of the hyper-sphere, approximation by a hyper-polyhedron. The polyhedrons are made of sides rotated by 45 degrees, which makes a
regular octagon in 2D case, or 18-sided polyhedron in 3D case. While the proposed modification slows the approach down in the trimming phase, as more trimming along additional (rotated) axes is to be performed, it significantly speeds up the exhaustive search phase. In the 9-dimensional space, the volume of the polyhedron is more than 10 times smaller than the volume of the corresponding hypercube.

**Randomized Trees by Lepetit et al.**

Lepetit et al. in [LLF05] formulate the matching as a classification problem. During training, anchor points on an image of a database object are constructed and described by pixel values in surrounding square patches. Each keypoint forms a separate class. To facilitate training, several training examples of each class (possible appearance of the patch around the keypoint location) are required. This is achieved by synthesising additional training views of the object using computer graphics rendering. The keypoints are extracted from each view independently, and only stable points, which were extracted in multiple views, are kept.

A classifier is learned which, during recognition, assigns to each query patch its corresponding class label, i.e. the corresponding database patch. An additional class is created for all query patches that do not match any database patch. Once the tentative correspondences are established, a standard RANSAC based method estimates the 3D pose. The classifier is implemented as a binary decision tree. Each non-terminal node of the tree contains a simple test comparing intensities of two pixels, that splits the image space into two parts. Each leaf contains an estimate, based on training data, of the conditional probability distribution over the classes, given that a query patch reaches that leaf. A query patch is classified by traversing the tree, and attributing it the class with the maximal conditional probability stored in the leaf it reaches.

The trees are constructed in top-down manner, where the tests are chosen by a greedy algorithm to best separate the given training examples. The process is recursively applied for descendant nodes, using only the database patches falling to that nodes. The recursion is stopped when a node receives too few patches, or when it reaches a specified depth. Since the number of classes, number of training examples and number of possible node tests are large, building one optimal tree is intractable. Instead, multiple randomised trees are used. For each tree, a small random subset of training examples and only a limited random selection of tests at each node is considered.

The method is demonstrated on real-time pose estimation of a single object. How it will scale to multiple objects is yet to be shown.

**Vocabulary Tree by Nistér**

David Nistér et al. [NS06] propose to use a vocabulary tree to achieve a real-time recognition of objects from large datasets. The approach is demonstrated on an impressive database of 50000 images. For feature extraction, they use the Maximally Stable Extremal Regions (MSERs) [MCUP04] in combination with Lowe’s SIFT descriptor [Lowe04]. Normalized SIFT descriptors are quantised with a vocabulary tree. The vocabulary tree defines a hierarchical quantisation that is built by hierarchical k-means clustering, k defines the branching factor (number of children of each node) of the tree. An initial k-means process is run on all training descriptors, giving k cluster centres. The training data are then partitioned into k groups according to proximity of descriptor vectors to a particular cluster centre. The process is then recursively repeated for each group of descriptors, splitting each quantisation group into k new parts.

A hierarchical scoring scheme is applied to rank database images, i.e. the score is affected also by non-leaf nodes which a query descriptor traverses on the path from the tree root to its
corresponding leaf. The correspondence is given only on per-image basis, the correspondence between individual query and database features is not established. Therefore, there can be no verification step involving global consistency of the matched features.

**Decision Trees on Random Subwindows by Marée et al.**

Marée et al. [MGPW05] has experimented with various decision tree based methods for classification of randomly extracted image windows. Their method extracts from training images a large number (about 120000) of possibly overlapping square subwindows of random sizes and at random positions. The same random process is applied to test images, only the number of subwindows is smaller (about 100). All subwindows are resized to a fixed scale (16 × 16 pixels) and transformed to a HSV colour space, i.e. represented by a vector of 768 numbers.

Five different tree-based classifiers were learned to classify the subwindows, four of them used a combination of multiple decision trees. Each tree outputs conditional class probability estimates for each subwindow. If multiple trees are used, all the predictions are averaged and the class corresponding to the largest aggregated probability is assigned to the query. Interesting recognition rates are obtained, especially considering that no geometric consistency was enforced.

**Literature on Evaluation of Interest Point Detectors and Descriptors**

Several papers emerged recently, where different interest-point detectors are evaluated and compared in performance to others.

In [SMB00, SMB98] Schmid et al. evaluate detectors by two criteria, repeatability and information content. Repeatability compares the geometrical stability of the detected points between different images of a given scene taken under varying viewing conditions. A point is ‘repeated’ if the 3D scene point detected in the first image is also accurately detected in the second one. Information content is a measure of the distinctiveness of an interest point. Distinctiveness is based on the likelihood of a local greyvalue descriptor computed at the point within the population of all observed descriptors. The criteria are designed to measure the quality of the interest points for tasks like image matching, object recognition and 3D scene reconstruction.

Five interest point detectors are compared, the Harris corner detector [HS88], Cottier, Horand [HVS90], Heitger [HRvdH+92] and Forstner [For94] detectors. The repeatability rates are evaluated under different conditions: image rotation, scale change, variable illumination, changing viewpoint and camera noise. The conclusion is that under all of the tested conditions the Harris corner detector performs equivalently or better than the other detectors.

Hall, Liebe and Schiele in [HLS02] discuss the quality of interest point detectors also with respect to the saliency of descriptions of the points’ local neighbourhoods. The saliency is defined to be inversely proportional to the probability of occurrence of that description. For the evaluation purposes, description by colour gaussian derivatives is used. Three detectors are evaluated, the Harris corner detector [HS88], Lindeberg’s scale-space interest point detector [Lin98], and the Harris-Laplacian interest point detector proposed by Mikolajczyk and Schmid in [MS01]. They conclude that all three interest point detectors are about equally suitable to provide a solution to the problem of image matching under scale changes. Since the Lindeberg and the Harris-Laplacian interest point detector require significant computation effort, they recommend to use the Harris detector.

In [SL01, STL+02], Sebe et al. compare wavelet-based interest point detector, described in [SL00], with the Harris corner detector. Again, two criteria were considered: repeatability rate and information content. The repeatability rate again evaluates the geometric stability of points
under different image transformations, the information content measures the distinctiveness of greylevel pattern at an interest point. The local patterns are described by rotationally invariant combinations of derivatives. The detectors were evaluated under image rotation and scaling. The results have shown that the wavelet-based detectors performed better than the Harris detector in both criteria.

In [KM03], Mikolajczyk and Schmid compare different descriptors of local appearance. They compare SIFT descriptors [Low99], steerable filters [FA91], differential invariants [KvD87], complex filters [SZ02b], moment invariants [GMU96] and cross-correlation for different types of interest point detectors [HS88, Low99, MS01, MS02]. The stability of descriptors was evaluated under changing illumination, and under scaling, rotation, and affine transformations. They observe that the ranking of the descriptors does not depend on the point detector and that the SIFT descriptor performs best. Steerable filters are second and are recommended as a good choice given their low dimensionality.

Finally, the most comprehensive and most recent comparison of affine-covariant region detectors is given in [MTS05]. The compared detectors are the 'Harris-Affine' detector [MS02, MS04, SZ02b], the 'Hessian-Affine' detector [MS02, MS04], the 'maximally stable extremal region' (MSER) detector [MCUP04] (also described in Section 3.2.1), an edge-based region detector [TG99, TG04], an intensity extrema-based region detector [TVG00, TG99], and an entropy-based detector of salient regions [KZB04]. The six detectors are compared on data set including structured and textured scenes. The performance is evaluated with respect to different types of transformations – viewpoint changes, scale changes, illumination changes, blur and JPEG compression. Repeatability of the detectors is measured by comparing the overlap between detected regions and the ground truth, which is known for the test images. Authors conclude that there does not exist one detector which outperforms the other detectors for all scene types and all types of image transformations. The detectors are complementary, i.e., they extract regions with different properties. Several detectors should be thus used simultaneously to obtain the best performance.
3 Local Affine Frames

An object or scene does never appear identical in two images. The cause is in the combined effects of environment conditions (e.g. illumination), camera and digitiser settings (e.g. gain, shutter speed, aperture, noise), camera projection parameters (e.g. focal length), and scene configuration (relative camera and object placement). Unless these conditions are well controlled, it is impossible to directly exploit raw image data to determine whether two images depict the same objects. Figure 3.1 shows two images of a cup. Even with a rather small change of the viewpoint and with constant illumination conditions, the pixel-wise differences of the object’s appearance are substantial. To recognise objects in images, it is necessary to find a better object representation than raw images; a representation that is invariant, or to a large degree robust, to possible variations of the appearance, but still discriminative enough to distinguish between different objects.

The approach pursued in this work is to describe the object’s appearance by a set of local patches. To facilitate comparison, patches that cover a particular object surface, independently of the viewpoint, are sought. Therefore the location, size and shape of the patches are derived solely from image content by a process that is covariant with affine transformation of the image. Photometrically invariant representations of the patches are then compared to establish patch-to-patch correspondences.

This chapter details how the affine geometric invariance is achieved. The invariance to photometric variations is described later in Section 4.1. In the first part of this chapter, Maximally Stable Distinguished Regions (MSERs) are defined and described. In the second part, an overview and a classification of affine-covariant constructions of local coordinate systems (frames) are presented, the affine covariance of the constructions is proven, and details on their computation are given. Finally, a technique to avoid generating an unnecessarily large number of frames is proposed. Frames are ordered by their added value to the object representation, and least valuable frames are discarded. The process maintains a representative coverage of the image, i.e. frames are first removed from over-represented locations.

3.1 Geometric Invariance

Let us assume that objects are observed by an approximately ideal pinhole (or central-projection, or perspective) camera. Radial (barrel) or any other non-linear distortion is neglected. Under
the pinhole camera model, the projections of a plane in the 3D scene under different views are related by linear projective transformations (homographies). The geometric invariance is achieved by detecting image-dependent Local Affine Frames (LAFs), i.e. local, affine, object-centric coordinate systems that “stick” to the image intensity profile and “follow” it as the image is deformed.

Why Local?

For a 3D object of generic shape, a change of viewpoint may induce an arbitrarily complex transformation of how the object appears in the image plane. The situation in Figure 3.1 is fairly simple, the depicted cup can be modelled as a cylinder. Global geometric transformation between views can be parameterised and accounted for during recognition. This is not generally so. Consider the image pair in Figure 3.2, there is no simple parameterisation of the deformation at global scale.

Let us then make a second assumption: any object surface can be reasonably well approximated as piece-wise planar. Even if an object has no planar parts at all, as in Figure 3.1, a piece-wise planar approximation can be made (e.g. by a 50-sided prism), such that the visual difference in the approximated appearance is small. It is not our goal to find this approximation; the proposition is only that treating local pieces of the object’s surface as planar does not induce large appearance differences. Under these two assumptions the change in object appearance, although complex globally, can be broken down to a set of homographies in the image plane.

![Figure 3.2: Another instance of the ICCV03 logo (right). There is no simple geometric transformation between the images due to non-rigid deformation of the subject.](image)

Why Affine?

Under the assumptions stated above, the most general group of (local) transformations to be considered are the homographies. Elastic, or even non-continuous, deformations are assumed not to happen at the local scale. Let us first review the hierarchy of plane-to-plane transformations, beginning with homographies:

**Linear projective transformation.** A linear projective transformation between two planes (a homography) is a map $F : \mathbb{R}^3 \to \mathbb{R}^3$ of the form $F(x) = Hx$, for all $x \in \mathbb{R}^3$, where $H$ is a $3 \times 3$ linear transformation of $\mathbb{R}^3$ and $x$ is a representation of a 2D point in homogeneous coordinates. Projections possess 8 degrees of freedom which arise from the nine elements...
3.1 Geometric Invariance

of the linear transformation, minus an overall scale factor which is not relevant in homogeneous representation. Projective transformation maintains concurrency and collinearity of points, tangency, inflections, and cross-ratio (ratio of ratio of lengths).

**Affine transformation.** The affine transformation of a plane is a map $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ of the form $F(x) = Mx + t$, for all $x \in \mathbb{R}^2$, where $M$ is a linear transformation of $\mathbb{R}^2$. In homogeneous coordinates, affine transformations are represented by matrices of the form

$$
A = \begin{pmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
0 & 0 & a_{33}
\end{pmatrix}
$$

The overall scale is again irrelevant, affine transformations thus have 6 degrees of freedom. Affine transformations also maintain the properties maintained by projective transforms and additionally they preserve parallelism of lines, the ratio of distances on a line or on parallel lines, and the ratio of areas.

**Similarity.** Similarity is a transformation that maintains all corresponding angles, and all distances are increased or decreased in the same ratio. In homogenous coordinates similarities are represented by

$$
S = \begin{pmatrix}
s r_{11} & s r_{12} & t_{13} \\
s r_{21} & s r_{22} & t_{23} \\
0 & 0 & 1
\end{pmatrix}
$$

where the upper $2 \times 2$ submatrix of $S$ is a 2D rotation matrix $R$ multiplied by a scale factor $s$, and $(t_{13}, t_{23})$ represents a translation vector. A similarity has 4 degrees of freedom (corresponding to the rotation angle, the scale factor, and the x- and y- displacements). Similarity transformations maintain, on top of the properties maintained by affine transformations, angles, curvatures and ratios of lengths.

**Euclidean transformation.** An euclidean (or orthonormal) transformation represents a rigid 2D motion applied to an object; a rigid motion consists of a translation and a rotation. The matrix representation in homogeneous representation is of the form

$$
E = \begin{pmatrix}
 r_{11} & r_{12} & t_{13} \\
r_{21} & r_{22} & t_{23} \\
0 & 0 & 1
\end{pmatrix}
$$

where the upper $2 \times 2$ submatrix of $E$ is again a 2D rotation matrix, and $(t_{13}, t_{23})$ is the translation vector. Euclidean transformations have 3 degrees of freedom, and additionally maintain distances and areas.

**Translation.** Finally, translation is a transformation consisting only of a constant offset, with no rotation or shape deformation. In homogeneous coordinates,

$$
T = \begin{pmatrix}
1 & 0 & t_{13} \\
0 & 1 & t_{23} \\
0 & 0 & 1
\end{pmatrix}
$$

where $(t_{13}, t_{23})$ is the offset. Translations have only 2 degrees of freedom, and additionally maintain absolute orientations.

Figure 3.3 shows local image patches, taken from images in Figure 3.1, aligned by transformations from aforementioned groups. The simplest model, translation, corresponds to the once
Figure 3.3: Image alignment under different transformation groups.
3.2 Distinguished Regions

The problem of object recognition was formulated in terms of finding reliable correspondences between images of a scene and of a database object. Local image elements that can be detected independently of affine transformations of the image, that are highly repeatable under varying viewpoint and illumination, and that are robust to image noise, are sought.

In most images there are regions that can be detected with high repeatability since they possess some distinguishing, invariant and stable property. Such regions of, in general, data-dependent shape, called distinguished regions (DRs) here, may serve as the elements to be put into correspondence. In this work, Maximally Stable Extremal Regions (MSERs) introduced in [MCUP04], are exploited.

An extremal region is a connected component of pixels which are all brighter (MSER+) or darker (MSER−) than all pixels on the region’s boundary. Extremal regions have two desirable properties. They are closed under continuous (and thus perspective) transformations of image coordinates, and they are closed under monotonic transformations of image intensities. An efficient (with near-linear complexity) and practically fast detection algorithm is outlined for a stable subset of extremal regions, the maximally stable extremal regions (MSERs).

3.2.1 Maximally Stable Extremal Regions (MSERs)

The Maximally Stable Extremal Regions are defined solely by an extremal property of the intensity function in the region and on its outer boundary. The concept can be explained informally as follows: Imagine all possible thresholdings of a grey-level image I. We will refer to the pixels below a threshold as “black” and to those above or equal as “white”. If we were shown a movie of thresholded images It, with frame t corresponding to threshold t, we would see first a white image. Subsequently black spots corresponding to local intensity minima would appear and grow. At some point regions corresponding to two local minima would merge. Finally, the last frame would be black. The set of all connected components in all frames of the movie is
Local Affine Frames

the set of all maximal regions; minimal regions could be obtained by inverting the intensity of \( I \) and running the same process. A maximal region is considered stable, if its area does not change significantly for a range of thresholds. Formal definition of the MSER concept is given in [MCUP04].

For certain regions, the local binarisation by thresholding is stable over a large range of thresholds. Such regions are of interest, since they possess the following properties:

- **Invariance to affine transformation of pixel values (image intensities).**
- **Covariance with adjacency-preserving** (continuous) transformations on the image domain.
- **Stability**, since only extremal regions whose support is virtually unchanged over a range of thresholds is selected.
- **Multi-scale detection.** Since no smoothing is involved, both very fine and very large structures are detected.
- **The set of all extremal regions can be enumerated in** \( O(n \log \log n) \), where \( n \) is the number of pixels in the image.

Enumeration of extremal regions proceeds as follows: First, pixels are sorted by intensity. The computational complexity of this step is \( O(n) \) if the range of image values is small, e.g. the typical \( \{0, \ldots, 255\} \), since the sort can be implemented as BINSORT [Sed88]. After sorting, pixels are placed in the image (either in decreasing or increasing order) and the list of connected components and their areas is maintained using the efficient union-find algorithm [Sed88], with complexity \( O(n \log \log n) \), i.e. almost linear.

The process produces a data structure storing the area of each connected component as a function of intensity. Intensity levels that are local minima of the rate of change of the area function are selected as thresholds producing maximally stable extremal regions. MSERs are uniquely identified by the position of a local intensity minimum (or maximum) and a threshold. The detection of a MSER is related to thresholding. Every extremal region is a connected component of a thresholded image. However, no global or “optimal” threshold is sought, all thresholds are tested and the stability of the connected components evaluated. The output of the MSER detector is not a binarised image. For some parts of the image, multiple stable thresholds exist. A system of nested subsets is output in this case.

Figure 3.4 shows an example of Maximally Stable Extremal Regions. Despite the surface deformations and viewpoint variations, a large number of regions cover the same area of the ICCV03 logo. The MSER detector exhibits high repeatability and robustness (shown experimentally in Section 6.2.1), which is benign for the object recognition task. As shown in Section 6.2.1, the repeatability and stability is higher for regions where the area remains unchanged for a large range of thresholds. In our experiments, we typically consider only regions which remain virtually unchanged for more than 12 consecutive thresholds.

3.2.2 Ordering of Image Pixels

The MSER algorithm requires an ordering on the image pixels, i.e. the algorithm works with scalar images. Having a colour image, how to order the RGB values? The common choice is to order the pixels by intensity, i.e. to convert the image to greyscale. Yet in many cases, depending on the actual problem being solved, different orderings can lead to better recognition. As an example, let us consider the traffic signs depicted in first row of Figure 3.5. The outer red rims are important regions for the sign detection. In the intensity projection (second row of
Figure 3.4: Examples of detected regions of MSER type. (a) input images, (b) MSER regions: MSER\(^-\) shown in red, MSER\(^+\) shown in blue.
Figure 3.5), the rims are not very distinctive. They are not even extremal regions, as they adjoin both brighter image pixels (the pole) and darker pixels (the trees), and are not therefore detected by the MSER detector. But if the RGB pixels are, for example, projected onto the red-blue axis of the RGB space, the rims become extremal and well separated from their neighbourhood (third row of Figure 3.5).

Figure 3.5 shows the results of projection of RGB values onto different axes of the RGB space. Original colour images are shown in the first row. The projection to intensity, i.e. to the black-white axis (second row), can be expressed as an inner product $s = (r, g, b) (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, where $(r, g, b)$ is a vector of RGB values of a pixel, and $s$ is the resulting scalar value. The remaining two rows in Figure 3.5 show projections onto the red-blue, $s = (r, g, b) (\frac{1}{2}, 0, -\frac{1}{2})^T$, and the green-magenta, $s = (r, g, b) (\frac{1}{4}, -\frac{1}{2}, \frac{1}{4})^T$ axes. These were chosen as they separate well certain image regions (make them extremal), and together with the intensity axis they form an orthonormal basis of the RGB space.

Figure 3.6 shows examples of non-linear RGB projections. Images in the first row depict saturations of the image colours, defined as $s = \sqrt{(r-I)^2 + (g-I)^2 + (b-I)^2}$, where $I$ is the intensity $I = \frac{r+g+b}{3}$. The remaining three rows show projections to intensity-normalised R, G, B channels respectively, where $s = \frac{r}{r+g+b}$, resp. $s = \frac{g}{r+g+b}$, resp. $s = \frac{b}{r+g+b}$.

Not only MSER detector, but any detector which process greyscale images, e.g. the Harris interest point detector, may benefit from pixel ordering different than by intensity. The choice of suitable ordering depends on actual application, namely on the nature of objects of interest. Traffic signs, for example, consist largely of red, blue and white regions, and some black and yellow. Using two RGB orderings, by intensity and by projecting onto red-blue axis, all the important regions are obtained with high confidence. If there is no apriori knowledge about which of the orderings facilitate the region detection best, multiple orderings are used simultaneously, which is computationally costly. The number of orderings can be reduced at run-time, automatically adapting to actual run-time conditions.

### 3.2.3 Adaptation to Environmental Conditions

The performance of the correspondence-search algorithm is increased and robustified when multiple constructions of distinguished regions are employed, e.g. when MSERs are computed using multiple orderings of image pixels concurrently. Although the number of used region constructions can be increased almost unlimitedly, the added value of further constructions is decreasing, and, inevitably, the computation speed is adversely affected. Moreover, depending on the scene, a large number of constructions might not be necessary in the first place. The point is to use only such subset of region constructions, that is essential for establishing the correspondences. Without imposing assumptions about the objects, scene, or the imaging system, the usefulness of a construction can be only roughly estimated from statistics gathered on general scenes. But in case of an environment stable in time, more precise estimation can be obtained by a run-time assessment, allowing a run-time adaptation of the method.

The adaptation is formulated as an optimisation problem. Assuming that the algorithm consecutively processes images from the same environment, the expected computation time is minimised, subject to the condition that the correspondence is still established. The cost function, which is minimised, the expected computation time, depends on (a) number of processes run to generate the regions, and (b) parameters of each process that control the amount of regions generated, e.g. a required region stability. The fulfilment of the condition of established correspondence depends on the task being solved. For the wide-baseline stereo problems, the solution is typically found if (a) a small absolute number (15–20) of correct local correspondences is found, and (b) if the percentage of correct correspondences is high enough (10–20%) to allow RANSAC or other robust estimator to identify correct epipolar geometry. In the object
3.2 Distinguished Regions

Colour image

Intensity

R-B projection

G-M projection

Figure 3.5: Examples of ordering of colour pixels by projection onto different axes in the RGB space. The rightmost column shows MSER regions detected in the transformed traffic sign image: MSER (red) and MSER+ (blue).
Figure 3.6: Examples of ordering of colour pixels by non-linear RGB transformations. The right-most column shows MSER regions detected in the transformed traffic sign image: MSER⁻ (red) and MSER⁺ (blue).
3.3 Geometric Primitives Covariant with Affine Transformations

A two-dimensional affine transformation possesses six degrees of freedom. Thus, to determine an affine transformation, six independent constraints, e.g. given by a correspondence of three non-collinear points, have to be found. The constraints are derived from various affine-covariant geometric primitives detected on image regions of sufficiently complex shape. In particular, we use directions (providing a single constraint), 2D positions (providing two constraints), and the covariance matrix of a 2D shape (providing three constraints).

Figure 3.8 presents an overview of the affine-covariant primitives. From regions output by a detector (left top corner), other regions are affine-invariantly derived (rectangular boxes). Indi-
1: Intensity, negative extrema
2: Intensity, positive extrema
3: Saturation, negative extrema
4: Saturation, positive extrema
5: Normalised red, negative extrema
6: Normalised red, positive extrema
7: Normalised green, negative extrema
8: Normalised green, positive extrema
9: Normalised blue, negative extrema
10: Normalised blue, positive extrema
11: Red-Blue projection, negative extrema
12: Red-Blue projection, positive extrema
13: Green-Magenta projection, negative extrema
14: Green-Magenta projection, positive extrema

Figure 3.7: Statistics gathered on different video sequences. Distribution of regions contributing to object detection according to pixel ordering.
3.3 Geometric Primitives Covariant with Affine Transformations

Individual primitives (elliptical boxes) are then computed, the flow of the computation is indicated by arrows. We divide the primitives into three categories:

Constructions derived from region shape only. The centre of gravity $\mu$ (Figure 3.8 (i)) of a region (the vector of first order algebraic moments) provides two constraints, e.g. resolves translation. The symmetric $2 \times 2$ covariance matrix $\Sigma$ (ii), the matrix of second central algebraic moments, gives 3 constraints. Together, the centre of gravity and the covariance matrix fix the affine transformation up to an unknown rotation. Normalisation by the covariance matrix (see Figure 3.10) therefore allows for affine-invariant measurement of distances, angles and curvatures. From these we derive the points of extremal distance to the centre of gravity (iii) (2 constraints) and the points of curvature extrema (iv) (2 constraints).

Another group of shape-derived primitives is obtained on concavities (v) (4 constraints for the two tangent points). Given a bitangent, the point on the region boundary farthest from the bitangent line (vi) is defined affine-covariantly (2 constraints, see Figure 3.12). A significant property of bitangents is their locality, i.e. they do not depend on correct detection of the whole region. If, for example, two regions get connected due to discretisation in one of the images, constructions based on integral characteristics, as is the centre of gravity or the covariance matrix, are incorrect, while concavities may be unaffected.

Figure 3.8: Overview of affine-covariant primitives. Rectangular blocks represent regions, detected or derived, and elliptical blocks represent the primitives. The numbering refers to Sections 3.3, 3.3.2, and to Fig. 3.13 Local affine frames are constructed by combining the primitives.
planar region $\Omega$ is a contiguous subset of $\mathbb{R}^2$.

affine transformation is a map $F : \mathbb{R}^n \to \mathbb{R}^n$ of the form $F(x) = A^T x + t$, for all $x \in \mathbb{R}^n$, where $A$ is a linear transformation of $\mathbb{R}^n$, assumed non-singular.

centre of gravity $\mu$ of a region $\Omega$ is $\mu = \frac{1}{|\Omega|} \int_\Omega x \, d\Omega$, where $|\Omega|$ is the area of the region.

covariance matrix of a region $\Omega$ is a $2 \times 2$ matrix defined as $\Sigma = \frac{1}{|\Omega|} \int_\Omega (x - \mu)(x - \mu)^T \, d\Omega$.

convex hull of a geometric object (such as a point set or a polygonal region) is the smallest convex set $S$ containing that object. A set $S$ is convex if whenever two points $P$ and $Q$ are inside $S$, then the whole line segment $PQ$ is also in $S$, or, equivalently, a set $S$ is convex if it is exactly equal to the intersection of all the half planes containing it.

bitangent is a line that is tangent to a curve at two distinct points. Bitangents contain segments of the convex hull that bridge concavities.

curvature $\kappa$ of a planar curve is defined by $\kappa = \frac{d\Phi}{ds}$ where $\Phi$ is the tangential (or turning) angle, and $s$ is segment length. The curve is convex in areas of positive curvature and concave in areas of negative curvature.

inflection point is a point on a curve at which the sign of the curvature $\kappa$ (i.e. the convexity of the curve) changes.

Table 3.1: Definition of terms

Finally, we exploit points of curvature inflections (vii), i.e. points where the shape changes from concave to convex or vice-versa (2 constraints), straight line segments (viii) of the region boundary (1 constraint for direction, or 4 constraints for end-points – if stable), and third order algebraic moments [Hei04] (ix) (1 constraint).

Constructions derived from image intensities. Several constraints can be derived from pixel values inside a region or in its neighbourhood. After normalisation by the covariance matrix, directions based on orientations of gradients (x), obtained for example as peaks of gradient histogram [Low04], or the direction of dominant texture periodicity (xi), determine the unknown rotation. Extrema of $R$, $G$, $B$ components (xii), or of any scalar function of RGB values provide 2 constraints.

Constructions derived from topology of regions. Finally, mutual configurations of regions are considered, i.e. nested regions, neighbouring regions, holes and incident regions. Region concavities and holes can be considered as distinguished regions of their own, and the computation of all of the constructions can be recursively applied. On the other hand, neither holes nor concavities have to be considered as part of the region, i.e. a convex hull can be substituted for the region, without losing the affine invariance.

3.3.1 Proofs of Affine Covariance of LAF Constructions

In the following we show that the geometric primitives used to establish local affine frames are indeed covariant with affine transformations. In particular, we show how the area, centre of gravity, and covariance matrix of a region changes under affine transformations of the region, and that the properties of tangent points and of the farthest-from-a-line points are maintained.
3.3 Geometric Primitives Covariant with Affine Transformations

**Region area.** Let us consider a region $\Omega_1$, and its image $\Omega_2$ transformed by affine transformation $A$, where for each $x_1 \in \Omega_1$ is $x_2 = A^T x_1 + t \in \Omega_2$. The area of $\Omega_2$ is given as

$$|\Omega_2| = \int_{\Omega_2} d\Omega_2 = \int_{\Omega_1} |A| d\Omega_1$$

where $|A|$ is the determinant of $A$, and $|\Omega|$ is the area of region $\Omega$. The area of a transformed region equals to $|A|$ times the area of the original region.

**The center of gravity.** The center of gravity of region $\Omega$ is $\mu = \frac{1}{|\Omega|} \int_{\Omega} x d\Omega$. The relation between the centres of gravity of transformed regions is:

$$\mu_2 = \frac{1}{|\Omega_2|} \int_{\Omega_2} x_2 d\Omega_2 = \frac{1}{|\Omega_2|} \int_{\Omega_1} (A^T x_1 + t)|A| d\Omega_1$$

$$= A^T \frac{1}{|\Omega_1|} \int_{\Omega_1} x_1 d\Omega_1 + \frac{1}{|\Omega_1|} \int_{\Omega_1} t d\Omega_1$$

$$= A^T \mu_1 + t,$$  \hspace{1cm} (3.2)

the centre of gravity changes covariantly with the affine transform.

**The covariance matrix.** The covariance matrix $\Sigma$ of a region $\Omega$ is a 2x2 matrix defined as $\Sigma = \frac{1}{|\Omega|} \int_{\Omega} (x - \mu)(x - \mu)^T d\Omega$. Covariance matrix of a transformed region $\Omega_2$ is then

$$\Sigma_2 = \frac{1}{|\Omega_2|} \int_{\Omega_2} (x_2 - \mu_2)(x_2 - \mu_2)^T d\Omega_2$$

$$= \frac{1}{|\Omega_1|} \int_{\Omega_1} (A^T x_1 + t - (A^T \mu_1 + t))(A^T x_1 + t - (A^T \mu_1 + t))^T |A| d\Omega_1$$

$$= \frac{1}{|\Omega_1|} \int_{\Omega_1} (A^T(x_1 - \mu_1))(A^T(x_1 - \mu_1))^T d\Omega_1$$

$$= A^T \left( \frac{1}{|\Omega_1|} \int_{\Omega_1} (x_1 - \mu_1)(x_1 - \mu_1)^T d\Omega_1 \right) A$$

$$= A^T \Sigma_1 A$$  \hspace{1cm} (3.3)

Cholesky decomposition of a symmetric and positive-definite matrix $\Sigma$ is a factorisation $\Sigma = U^T U$, where $U$ is an upper triangular matrix. Cholesky decomposition is defined up to a rotation, since $U^T U = U^T R^T R U$ for any orthonormal $R$. For the decomposition of covariance matrix of a transformed region we write

$$\Sigma_2 = U_2^T R_2^T R_2 U_2 = A^T \Sigma_1 A$$

thus

$$U_2^T = A^T U_1^T R$$  \hspace{1cm} (3.4)

Hence the triangular matrix $U$, obtained as the cholesky decomposition of a covariance matrix $\Sigma$, is covariant, up to an arbitrary orthonormal matrix $R$, with the affine transform applied to the region.

**Line parallelism.** Let us consider two lines, determined by points $p$ and $q$, and $r$ and $s$ respectively. The lines are parallel, iff

$$(p - q) = k(r - s), \quad k \in \mathbb{R} \setminus \{0\}$$  \hspace{1cm} (3.6)
Affinely transformed lines are then parallel iff
\[
(A^T p + t - A^T q - t) = k(A^T r + t - A^T s - t)
\]
\[
(A^T p - A^T q) = k(A^T r - A^T s)
\]
\[
A^T (p - q) = kA^T (r - s)
\]
\[
(p - q) = k(r - s)
\]
which is true if and only if the lines were parallel before the transformation. Thus, affine transformation preserves line parallelism.

**Ordering of distances to a line:** Let us have a line determined by two points \( p \) and \( q \). For a point \( x \), its distance \( d_1 \) to the line \( pq \) is \( d_1 = \frac{2S}{pq} \), where \( S \) is the area of the \( pqx \) triangle. Using eq. 3.1, it follows that the transformed distance \( d_2 \) is given as
\[
d_2 = \frac{2|A||S|}{|A^T p + t - A^T q - t|} = \frac{|A||p - q|}{|A^T p - A^T q|} d_1 = k d_1
\]
where \( k \) is a scalar constant for given line \( pq \) and transformation \( A \). Affine transformation thus preserves ordering of distances between points and a line. It directly follows, that a point \( x \in X \) with the property of being of all the points in \( X \) the farthest one from line \( pq \), retains its property under affine transformations.

**Incidence of points and lines:** Under affine transformations, points incident with a line will remain on the line, and, vice-versa, distinct points will not be brought to the line unless the transformation is singular. The property is again easily shown exploiting the covariance of region area, from Equation. 3.1. Considering a line defined by two distinct points \( p \) and \( q \), and a point \( x \), the area \( S_1 \) of the \( pqx \) triangle equals to zero if \( x \) is on \( pq \) and nonzero otherwise. After affine transformation, the area of the triangle becomes \( S_2 = |A|S_1 \), where \( |A| \) is the determinant of the transformation matrix (\( S_2 \) is the area of triangle given by points defining the transformed line, i.e. \( A^T p + t \) and \( A^T q + t \), and the transformed point \( A^T x + t \)). Assuming non-singular transformation, i.e. \( |A| \neq 0 \), the transformed triangle has area \( S_2 = 0 \) if and only if \( S_1 = 0 \). Thus the incidence is maintained.

**Tangent and bitangent lines:** Tangent line is a line incident with region boundary (in a tangent point \( p \)), which does not pass through any of the region interior points. Since the incidence property between the tangent line and the boundary, respective interior points, is maintained, the line transformed by an affine transformation remains tangent to the transformed region, and the tangency occur in the transformed point \( p_2 = A^T p + t \). An analogy holds for the bitangent lines, where both tangent points are maintained.

An affine transformation is either orientation-preserving or orientation-reversing, according to whether determinant \(|A|\) is positive or negative respectively [MZ92]. Therefore the sign of the curvature \( \kappa = \frac{d\phi}{ds} \) is for a transformed region either reversed or preserved, consistently along its whole contour. It follows that linear segments of the contour (segments of zero curvature) and inflection points (points where the curvature changes its sign, without specifying whether from positive to negative or vice versa) are maintained.

### 3.3.2 Details on Detection of the Geometric Primitives

Details on the identification of the affine-covariant geometric primitives are given here. The process starts from a set of polygonal image regions (MSERs here, but any affine-covariant regions apply). Each region is first smoothed to reduce discretisation artefacts. Then the individual primitives are computed, such as the matrix of second moments, contour curvature extrema, inflection points, or bitangents.
3.3 Geometric Primitives Covariant with Affine Transformations

![Figure 3.9: Examples of detected MSER regions. (a) detected regions represented by a polygon consisting of pixel boundary segments, (b) the same regions after contour smoothing.](image)

**Preprocessing**

A region is a connected sets of image pixels. A polygonal representation is constructed from its outer boundary. As is visualised in Figure 3.9 (a), shape of the regions, especially of small regions, is severely affected by image rasterisation. To reduce the rasterisation artefacts, the polygons are first smoothed. The x- resp. y- coordinates of the polygon vertices are organised into a cyclic list and smoothed with a one dimensional Gaussian kernel [MM92]. To accommodate for scaling of the regions, the standard deviation $\sigma$ of the kernel depends on the region size, in our implementation

$$\sigma = \max\left(\frac{\sqrt{|\Omega|}}{k}, 1\right), \quad (3.9)$$

where $|\Omega|$ is the region area, i.e. the number of region pixels, and $k = 30$ is a parameter controlling the amount of the smoothing. Such a simple approach to smoothing is not affine-invariant, but is sufficiently suppressing the discretisation effects while preserving the region shape. Section 6.2.4 experimentally shows the impact of the smoothing, and of choice of the parameter $k$, on recognition rate. The regions are henceforth treated as simple (non-intersecting) polygons with non-integer coordinates, region holes are treated separately. Examples of smoothed contours are shown in Figure 3.9 (b). Alternatively, the regions could have been represented in a parametric form, by splines for example, which might have allowed sub-pixel localisation of important contour points.

**Detection of Individual Affine-Covariant Primitives**

Let us have a polygon $\Omega$ with $n$ vertices. Let us denote $x_i$ and $y_i$ the $x-$ and $y-$ coordinate of $i$th vertex respectively. The polygon is closed, so $x_0 = x_n, y_0 = y_n$. The algorithms for computation of region moments: area (zero order algebraic moment), centre of gravity (first order algebraic moments) and covariance matrix (second order central algebraic moments), follow the standard algorithm for computation of the area of a non-intersecting polygon, where the area is incrementally updated for vertical strips bounded by $x-$ coordinates of two neighbouring
vertices:

\[
\mu_{pq} = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}} (y - y_{i-1}) \frac{x - x_{i-1}}{x_{i} - x_{i-1}} x^p y^q \, dy \, dx,
\]

resp. \hspace{1cm} (3.10)

\[
\mu'_{pq} = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}} (y - y_{i-1}) \frac{x - x_{i-1}}{x_{i} - x_{i-1}} (x - \mu_{10}) (y - \mu_{01}) \, dy \, dx,
\]

where \( p, q \) denote the order of the moments in \( x - \) and \( y - \) respectively, \( \mu_{pq} \) denotes algebraic moment of order \( p, q \), and \( \mu'_{pq} \) denotes central algebraic moments. The area of region \( \Omega \) is computed as

\[
|\Omega| = \mu_{00} = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}} \frac{1}{\sqrt{\sigma_{11}}} \, dy \, dx
\]

\[
= \frac{1}{2} \sum_{i=1}^{n} (x_i - x_{i-1})(y_i + y_{i-1}) = \frac{1}{2} \sum_{i=1}^{n} (x_i y_{i-1} - x_{i-1} y_i)
\]

The centre of gravity is

\[
\mu = (\mu_{10}, \mu_{01})^T
\]

\[
\mu_{10} = \frac{1}{6|\Omega|} \sum_{i=1}^{n} (x_i - x_{i-1}) (2y_{i-1}x_{i-1} + y_ix_{i-1} + y_{i-1}x_i + 2y_i x_i)
\]

\[
= \frac{1}{6|\Omega|} \sum_{i=1}^{n} (x_i + x_{i-1})(y_{i-1}x_i - x_{i-1} y_i)
\]

\[
\mu_{01} = \frac{1}{6|\Omega|} \sum_{i=1}^{n} (x_i - x_{i-1}) (y_i^2 + y_i y_{i-1} + y_{i-1}^2)
\]

\[
= \frac{1}{6|\Omega|} \sum_{i=1}^{n} (y_i + y_{i-1})(y_{i-1}x_i - x_{i-1} y_i)
\]

For the covariance matrix (matrix of second central algebraic moments) we write:

\[
\Sigma = \begin{pmatrix}
\mu_{20} & \mu_{11} \\
\mu_{11} & \mu_{02}
\end{pmatrix}
\]

\[
\Sigma_{11} = \mu_{20} = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}} \frac{x' - x_{i-1}}{x_{i} - x_{i-1}} x'^2 y' \, dx' 
\]

\[
= \frac{1}{12|\Omega|} \sum_{i=1}^{n} (x_i - x_{i-1})
\]

\[
(3x_i^2 y_{i-1} + x_{i-1}^2 y_i + 2x_i x_{i-1} y_i + 2x_i x_{i-1} y_{i-1} + x_{i-1}^2 y_i + 3x_{i-1}^2 y_{i-1})
\]

\[
= \frac{1}{12|\Omega|} \sum_{i=1}^{n} (x_i y_{i-1} - x_{i-1} y_i)(x_i^2 + x_i x_{i-1} + x_{i-1}^2)
\]

(3.15)
The third central moments of a region are given by (omitting here the factorisation of the integrands):

\[\Sigma_{22}^i = \mu_{02}^i = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}+(y_{i}-y_{i-1})}\frac{x-x'_{i-1}}{x-x'_{i-1}} y^2 \, dy' \, dx' \]

\[= \frac{1}{12|\Omega|} \sum_{i=1}^{n} (x_i' - x_{i-1}) \left(y_i'^3 + y_i'^2 y_{i-1}' + y_i' y_{i-1}'^2 + y_{i-1}'^3\right)\]

\[= \frac{1}{12|\Omega|} \sum_{i=1}^{n} (x_i'y_{i-1}' - x_{i-1}'y_{i}')(y_i'^2 + y_i'y_{i-1}' + y_{i-1}'^2) \quad (3.16)\]

\[\Sigma_{12}^i = \mu_{11}^i = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}+(y_{i}-y_{i-1})}\frac{x-x'_{i-1}}{x-x'_{i-1}} x'y' \, dy' \, dx' \]

\[= \frac{1}{24|\Omega|} \sum_{i=1}^{n} (x_i' - x_{i-1}) \left(3x_i'y_i'^2 + x_{i-1}'y_{i}'^2 + 2x_i'y_{i-1}'y_i' + 2x_{i-1}'y_i'y_{i-1}' + x_{i-1}'y_{i-1}'^2 + 3x_i'y_{i-1}'^2\right)\]

\[= \frac{1}{24|\Omega|} \sum_{i=1}^{n} (x_i'y_{i-1}' - x_{i-1}'y_{i}')(2x_i'y_i' + x_{i-1}'y_{i}' + x_{i-1}'y_{i-1}' + 2x_{i-1}'y_{i-1}') \quad (3.17)\]

\[\Sigma_{21} = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}+(y_{i}-y_{i-1})}\frac{x-x'_{i-1}}{x-x'_{i-1}} y'x' \, dy' \, dx' \]

\[= \Sigma_{12} \quad \text{(3.17)}\]

where \(x' = x - \mu_{10}\) and \(y' = y - \mu_{01}\).

The third central moments of a region are given by (omitting here the factorisation of the integrands):

\[\mu_{30}^i = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}+(y_{i}-y_{i-1})}\frac{x-x'_{i-1}}{x-x'_{i-1}} x^3 \, dy \, dx \]

\[\mu_{21}^i = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}+(y_{i}-y_{i-1})}\frac{x-x'_{i-1}}{x-x'_{i-1}} x^2y' \, dy \, dx \]

\[\mu_{12}^i = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}+(y_{i}-y_{i-1})}\frac{x-x'_{i-1}}{x-x'_{i-1}} x'y^2 \, dy \, dx \]

\[\mu_{03}^i = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_{i}} \int_{y_{i-1}}^{y_{i}+(y_{i}-y_{i-1})}\frac{x-x'_{i-1}}{x-x'_{i-1}} y^3 \, dy \, dx \]

Once the covariance matrix is computed, the region shape is normalised so that the covariance matrix of the resulting shape equals to the identity matrix. This is achieved by transforming every polygon vertex by the inverse of Cholesky decomposition of the covariance matrix, i.e. by \(A = (\text{chol}(\Sigma))^{-1}\). The effect is illustrated in Figure 3.10, where a detected region (a) is transformed into its normalised shape (b).

Shape normalisation, together with the position of the centre of gravity of the region, fixes the affine transformation up to an unknown rotation. The orientation is determined in several ways. First, the orientation is given by direction from the centre of gravity to the contour points of extremal distance to the centre (iii). The distances are shown in Figure 3.10 (c). Vertices of extremal distance are obtained with the non-maxima suppression algorithm. Second, the orientation is obtained from local extrema of curvature of the shape-normalised contour (iv).
3 Local Affine Frames

Figure 3.10: Shape normalisation by the covariance matrix. (a) a detected region, (b) the region shape-normalised to have a unit covariance matrix, (c) distances to the centre of gravity, (d) curvature estimation, (e) curvature of the normalised shape.

The computation of the approximate curvature proceeds as follows: For each vertex \( x \), two segments \( l = x\tilde{x} \) and \( r = x\tilde{y} \) of length \( a \) are spanned in opposite directions along the polygon boundary (see Figure 3.10 (d)). The cosine of the angle \( \phi \) is

\[
\cos \phi = \frac{l_x r_x + l_y r_y}{|l||r|},
\]

from which the curvature \( \kappa \) is estimated as

\[
\text{curvature } \kappa = s \frac{1 + \cos \phi}{2}, \quad \text{where } s = \begin{cases} 1 & \text{if } l_x r_y - l_y r_x > 0 \\ -1 & \text{otherwise} \end{cases}
\]

(3.18)

The curvature \( \kappa \) ranges from \(-1\) to \(1\), equals to \(0\) for straight segments, and is negative for concave and positive for convex curvatures. An example of the curvature values is shown in Figure 3.10 (e). The segment length \( a \) controls the scale at which the curvature is computed. Since the regions are shape and scale normalised, \( a \) is of a fixed value and need not be adapted to individual regions. Figure 3.10 (e) shows curvatures computed for two different values of \( a \), \( a = 0.5 \) (blue line) and \( a = 0.2 \) (red line). In our implementation we use \( a = 0.5 \).

The region orientation is also resolved using inflection points (vii). Two segments of the length \( a \) are spanned from every polygon vertex. An inflection point is identified if all vertices under one of the segments have positive curvature and all vertices under the another one have negative curvature. Third algebraic moments (ix) of the region shape provide another way to determine the unknown orientation. Following the method described in [Hei04], the third moments of the shape-normalised region form a complex number

\[
c = \mu'_30 + \mu'_12 + i(\mu'_21 + \mu'_03),
\]

(3.19)

whose phase angle

\[
\alpha = \tan^{-1}\left(\frac{\mu'_{21} + \mu'_{03}}{\mu'_{30} + \mu'_{12}}\right)
\]

(3.20)

changes covariantly with the region orientation. Finally, the orientation is obtained from direction of straight linear segments on the region boundary (viii). A standard Douglas-Peucker algorithm [DP73, Ram72] is executed on the shape-normalised region. At each step, the algorithm attempts to approximate a sequence of vertices by a line segment connecting first and last vertices of the sequence. The sequence vertex farthest from the line segment is found, and if the distance is below a predefined threshold, the approximation is accepted. Otherwise, the algorithm is recursively applied to the two subsequences created by splitting the sequence at the farthest vertex. Since the algorithm is performed on shape-normalised contours, it is affine-invariant even if the involved threshold on maximal approximation error is fixed for all regions (we choose value 0.03). See Figure 3.11 for an illustration of the process.

Concavities (v) are identified with segments of the region boundary that depart from the convex hull of the region. For each concavity, two points are found maximising distance to the
corresponding bi-tangent line (vi). One of them is located on the contour segment that forms the concavity, the other one on the rest of the contour. Figure 3.12 illustrates a complex, non-convex region with six concavities. Figure 3.12 (a) shows the centre of gravity and the covariance matrix for each concavity, which are computed by an identical process as for the whole region. Figure 3.12 (b) demonstrates, for one of the concavities, the two points of maximal distance.

3.4 LAF Construction

A frame is constructed by combining affine-covariant primitives which constrain all of its six degrees of freedom. The combinations we used in the experiments are illustrated in Figure 3.13. The images show basis vectors of the frames along with the primitives – points (e.g. inflection points), linear segments (e.g. bitangents), and ellipses representing covariance matrices. Figure 3.13 includes a table listing, for each of the frame types, the combination of primitives that define it.

The list of frame constructions can be easily expanded by arbitrary combinations of the affine-covariantly detected primitives. Using multiple frames per a detected region yields highly redundant coverage the object appearance, which supports robust recognition. Typically, the average number of frames per a region is about eight. Figure 3.14 shows an example of how the frames might cover the images.
3 Local Affine Frames

Figure 3.13: Examples of local affine frames of different types. The table indicates which affine-covariant primitives were combined to obtain the frames.

* Affine-covariant localisation of curvature extrema requires prior shape normalisation by covariance matrix.
Figure 3.14: Coverage of images by local frames. (a) original query and database images, (b) image coverage by local coordinate systems, whiter area – more overlapping frames, (c) frames where correspondences between the images were found (including mismatches), (d) image area covered by corresponding frames.

3.4.1 Representation of Local Affine Frames

Local affine frames are represented as a matrix of the affine transformation which maps the canonical coordinate system into image coordinates. If, for example, the local frame is constructed from three detected image points \( p, q, r \), the result is a transformation that transforms points \((1,0)^T\), \((0,1)^T\) and \((0,0)^T\) to \( p, q, \) and \( q \) respectively. The transformation is considered in homogenous coordinates, so the matrix representation is in the form

\[
A = \begin{pmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
0 & 0 & 1
\end{pmatrix}
\]

In following we show how is the transformation deduced from different combinations of geometric primitives. First, suppose the case of three points \( p, q, r \), which might for example represent two tangent points of a concavity and the centre of gravity of the region (Figure 3.13 (e)). The affine transformation \( A \) is computed as

\[
A = \begin{pmatrix}
M_{21} & M_{22} & M_{23} \\
M_{31} & M_{32} & M_{33} \\
0 & 0 & 1
\end{pmatrix}^{-1}
, \quad \text{where} \quad M = \begin{pmatrix}
p_x & q_x & r_x \\
p_y & q_y & r_y \\
1 & 1 & 1
\end{pmatrix}^{-1}
\]

(3.21)

Another case is when we have the covariance matrix \( \Sigma \), a point \( p \) (e.g. the centre of gravity), and a direction given by a vector \( v \) (Figure 3.13 (h, j)). First, the shape-normalising transformation is obtained from the covariance matrix and the direction vector \( v \) is deskewed:

\[
M = \text{chol}(\Sigma), \quad \text{and} \quad u = M^{-1}v
\]

(3.22)

The angle of rotation \( \phi \) is determined and the shape normalisation is combined with the rotation:

\[
\phi = \tan^{-1}\left(\frac{u_y}{u_x}\right), \quad \text{and} \quad N = M \begin{pmatrix}
\cos(\phi) & -\sin(\phi) \\
\sin(\phi) & \cos(\phi)
\end{pmatrix}
\]

(3.23)
The transformation $A$ is then constructed as

$$
A = \begin{pmatrix}
N_{11} & N_{12} & px \\
N_{21} & N_{22} & py \\
0 & 0 & 1
\end{pmatrix}
$$

(3.24)

Finally let us consider the situation where we have the covariance matrix $\Sigma$ and two points $p$ and $q$, for example the centre of gravity and a point of extremal curvature (Figure 3.13 (b)). Since such configuration provides seven constraints, one is superfluous. Would we drop the distance between the two points, i.e. instead of the location of one of the point we would have only a direction, the case would be identical to the previous one. Here we drop the size of the region. The covariance matrix is used only to deskew the shape, but the scale is given by the distance of the two points. The first step is again to obtain the deskewed direction. The shape-normalising transformation $M$ now does not scale the objects:

$$
M = \frac{\text{chol}(\Sigma)}{\sqrt{|\Sigma|}}, \quad \text{and } u = M^{-1}(q - p)
$$

(3.25)

where $|\Sigma|$ is the determinant of $\Sigma$. The rotation angle $\phi$ is computed and the shape normalisation is combined with the rotation and scaling determined from the distance of the points $p$ and $q$.

$$
\phi = \tan^{-1}\left(\frac{u_y}{u_x}\right), \quad \text{and } N = \sqrt{u_x^2 + u_y^2} M \begin{pmatrix}
\cos(\phi) & -\sin(\phi) \\
\sin(\phi) & \cos(\phi)
\end{pmatrix}
$$

(3.26)

Finally, the transformation $A$ is constructed as in the previous case.

$$
A = \begin{pmatrix}
N_{11} & N_{12} & px \\
N_{21} & N_{22} & py \\
0 & 0 & 1
\end{pmatrix}
$$

(3.27)

### 3.5 Reduction of the Number of Generated LAFs

Since multiple frames are constructed for each detected region, a highly redundant representation of the object appearance is obtained. While this redundancy supports robust recognition, it also negatively affects memory requirements and recognition speed of the method. Moreover, if the matching score is based on the number of frames put into correspondence, the recognition decision is biased towards complex-shaped regions. The example in Figure 3.15(a) shows frames constructed on a part of an input image. As can be seen, the density of the frames depends on the complexity of the underlying region. The number of frames on regions of complex shape with many concavities, like the letter “M”, can be several orders higher then the number of frames on simple shapes, as is the letter “O”.

An adaptation approach similar to the one described in section 3.2.3 can be used to find the optimal subset of relevant LAF constructions. Unfortunately, the success of individual constructions in matching does not show significant dependence on the nature of objects being recognised. Approximately, neglecting one type of LAFs proportionally decreases the recognition rate. Better results are obtained if the frames are reduced across frame constructions, i.e. the frames are selected by other characteristics than the type of construction.

#### 3.5.1 Selection of LAFs Based on Uniform Image Coverage

A LAF reduction procedure is proposed, where the stress is put on maintaining a representative coverage of the image. Frames are first removed from over-represented locations. The approach
3.5 Reduction of the Number of Generated LAFs

Figure 3.15: Limiting the number of frames representing an image. (a) all constructed LAFs, their density depends on the complexity of the underlying region. (b) LAF set reduced to 50%, (c) 25%, (d) 10%, while retaining coverage of the image.

is asymmetric, database and query images are treated differently. The frames are reduced only in query images, while the database images are represented by all constructed frames. This way the hard part, that is, identifying identical subsets of frames on an object when viewed in different images, is circumvented. The task is therefore, given a set of query frames and their required reduced number, to select a maximally representative subset of the frames, a subset which well maintains recognition of objects of interest.

Algorithm 2 Reduction of the Number of LAFs while Maintaining Image Coverage

Input: A set $\mathcal{S}$ of detected LAFs, a required number $N$ of retained frames, or maximal overlap $O_{\text{max}}$

Output: A representative subset of $\mathcal{S}$ of cardinality $N$

1. Sort $\mathcal{S}$ by MSER stability in descending order
2. For each frame $F_i \in \mathcal{S}$ compute cumulative overlap $O_i$ with all more stable frames:
   $$O_i := \sum_{j<i} \frac{|\Omega_{F_i} \cap \Omega_{F_j}|}{|\Omega_{F_i} \cup \Omega_{F_j}|}$$
   where $\Omega$ is an image area corresponding to square region $(0,1) \times (0,1)$ in the frame coordinates
3. Sort $\mathcal{S}$ by overlap $O$ in ascending order
4. Return first $N$ elements of $\mathcal{S}$, or elements of $\mathcal{S}$ with overlap $O < O_{\text{max}}$

Let us motivate the LAF selection with the following observations:

- An object of interest can be small in the image, and arbitrary background can be found
outside the object. Therefore, to make the frame selection indifferent to the background, the selection has to be local (in image coordinates). Would we try to impose any global ordering of the frames, selecting then 'best' N frames in the image, it may lead to suppression of all of the frames representing the object. For any chosen global LAF ordering, a scene background can be found that will dominate the selection.

- The number of MSER regions on an object is unknown and highly varying. For some objects of interest, e.g. a product logos, the complete object can be covered by a single region of complex shape. Other objects might be covered by many simple-shaped regions. The LAF selection therefore should not be connected to underlying MSER regions, as e.g. in selecting 'best' N frames on each region, but rather directly on the spatial distribution of LAFs.

- The best LAF repeatability predictor found experimentally is the stability of its underlying MSER region. The stability of a MSER is given as the range of thresholds over which the region does not change (see Sections 3.2.1 and 6.2.1).

- Because of the asymmetric approach, it is not important which of competing frames, i.e. frames estimated similarly repeatable, is selected.

The observations lead to the formulation of the LAF reduction algorithm presented in Algorithm 2. Frame overlap is defined identically as the frame alignment in experiments in Section 6.2.2 (see Figure 6.10). The algorithm effectively orders the LAFs by the sum of overlaps with all LAFs with higher stability, and retains frames with the sum lowest. The cardinality N of the retained frame set is monotonously related to the maximal cumulated overlap $O_{\text{max}}$. Either, or both, can be used to control the algorithm output. Specifying the cardinality $N$ limits processing time of scenes with many frames. Specifying the maximal overlap $O_{\text{max}}$ assures uniform LAF coverage across all query images, not only over a single one.

The algorithm outcome is illustrated in Figure 3.15. In (a), all the constructed frames are shown. The remaining figures show reduced frame sets, with the number of retained frames approximately 50% (b), 25% (c), and 10% (d) of the original number. As desired, LAFs are first removed from redundantly represented areas (complex-shaped letters “M”, “P”, and “C”), while the single frame representing the letter “O” remains. The effect of the reduction on both the recognition rate and the recognition speed is experimentally verified in section 6.2.3.
4 Local Correspondences

Recognition of objects, which are represented by a single training image, requires invariance to large differences in viewpoint and in environmental conditions, e.g. illumination. The process of establishing of correspondences can be made invariant at two stages. Either an invariant representation is built for each image, and the similarity measure, used to identify similar representations, has then a simple form, e.g. euclidean or mahalanobis distance. Or the invariance is achieved during evaluation of the similarity, which is then computationally complex. The disadvantage of the later approach is that the complex evaluation of the similarity effectively prevents us from using an indexing scheme of any kind. It is therefore hard, if not impossible, to establish correspondences with sublinear time complexity.

Since we find fast retrieval an important aspect of the recognition system (a decision-tree based approach for sublinear retrieval is proposed later in Chapter 5), our course is to build the geometrically and photometrically invariant representation of objects. Previous chapter detailed constructions of local affine-covariant coordinate systems that are fully defined by image measurements. As such, they “stick” to the objects in the image if the viewpoint changes, and serve as object-centred frames of reference, providing thus theoretical invariance and practical stability under varying viewpoint. This chapter continues with description of illuminant-invariant representation of local appearance and of the process of formation of local correspondences.

4.1 Normalisation of Measurement Region

Invariance of the object representation to geometric variations is achieved by normalising local appearance according to the detected frames. Image neighbourhood of every LAF is transformed into a canonical coordinate system, and a geometrically normalised patch is constructed. The patch is then normalised photometrically.

**Geometric normalisation.** The affine transformation between the canonical frame with origin \( O = (0,0)^T \) and basis vectors \( e_1 = (1,0)^T \) and \( e_2 = (0,1)^T \) and a frame established in the image is described in homogenous coordinates by a 3 by 3 matrix (see Section 3.4.1)

\[
A = \begin{pmatrix}
    a_1 & a_2 & a_3 \\
    a_4 & a_5 & a_6 \\
    0 & 0 & 1
\end{pmatrix}
\]

**Measurement region (MR)** is the part of the image, defined in terms of the affine frame, whose appearance, after appropriate encoding (see Section 4.2), is used to determine local correspondences. Each local affine frame is associated with one or possibly multiple MRs. The choice of MR shape and size is arbitrary. Larger MRs have higher discriminative potential, but are more likely to cover part of an object that is not locally planar, or a part of the image outside the object. The MR shape is related to the type of used descriptor and is usually square (e.g. for SIFT descriptor [Low04]) or circular (e.g. for ShapeContext [BMP00] which represents the images in polar coordinates). Measurement regions can be even of variable, data-driven shape or size, but then the problem with indexing arises again.

The SIFT and DCT (described later) descriptors which we employ require square MRs. Based on experimental evaluation (Section 6.2.6), the size of the region is set to \((-1,2) \times (-1,2)\) in the frame coordinate system. In image coordinate system, the measurement region of a
frame represented by transformation matrix $A$ becomes a parallelogram with corners at (in homogeneous coordinates):

$$c_1 = A \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}, \quad c_2 = A \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix}, \quad c_3 = A \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}, \quad c_4 = A \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}.$$ 

Pixels from the parallelogram in image $I$ are transformed by $A^{-1}$ to a geometrically normalised patch $I'$. The resolution of the patch is experimentally established in Section 6.2.7.

**Photometric Normalisation.** A linear camera (i.e. a camera without gamma-correction) is assumed and specular reflections and shadows are ignored. The combined effect of different scene illumination and camera and digitiser settings (gain, shutter speed, aperture) is modelled by affine transformations of individual colour channels, leading to the photometric transformation between two corresponding database and query patches $I'^D$ and $I'^Q$ in the form:

$$\begin{pmatrix} r_Q \\ g_Q \\ b_Q \end{pmatrix} = \begin{pmatrix} m_r & 0 & 0 \\ 0 & m_g & 0 \\ 0 & 0 & m_b \end{pmatrix} \begin{pmatrix} r_D \\ g_D \\ b_D \end{pmatrix} + \begin{pmatrix} n_r \\ n_g \\ n_b \end{pmatrix} \quad (4.1)$$

The parameters $m_r, n_r, m_g, n_g, m_b, n_b$ differ for individual corresponding patches. This model agrees with the monochromatic reflectance model [Hea89] in the case of narrow-band sensor. It can be viewed as an affine extension of the diagonal model that has been shown by Finlayson to be sufficient in common circumstances [FDB94], at least in conjunction with sensor sharpening [DFB94]. To represent a patch $I'$ invariantly to photometric transformations, intensities are transformed into a canonical form $\hat{I}'$. The intensities of individual colour channels are affinely transformed to have zero mean and unit variance. A photometric transformation $P$, which recovers the original image colours from the normalised form, is

$$P = \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_g & 0 \\ 0 & 0 & \sigma_b \end{pmatrix} + \begin{pmatrix} \mu_r \\ \mu_g \\ \mu_b \end{pmatrix}, \quad I'(x) = P\hat{I}'(x), \quad (4.2)$$

where $\mu_{r,g,b}$ are mean values and $\sigma_{r,g,b}$ are standard deviations of individual colour channels of patch $I'$. The diagonal form of the photometric transformation suffices for the purpose of establishing of correspondences. A more complex form, e.g. linear or affine, can be obtained in special cases, as shown in Chapter 7.

The normalisation procedure of a local patch is summarised in Algorithm 3. The twelve normalisation parameters ($a_1 \ldots a_6$ for geometric normalisation, $\sigma_r, \mu_r, \sigma_g, \mu_g, \sigma_b$ and $\mu_b$ for photometric normalisation) are stored along with the descriptor of the normalised local patch.

**Algorithm 3 Normalisation of a Local Representation**

1. Establish a local affine frame, form the affine transformation $A$ between canonical coordinate system and the detected frame.

2. Express the intensities of the $A$’s measurement region in the canonical coordinate system $I'(x) = I(Ax), \quad x \in \text{MR}$ with some discretisation.

3. Apply the photometric normalisation $\hat{I}'(x)_{r,g,b} = (I'(x)_{r,g,b} - \mu_{r,g,b})/\sigma_{r,g,b}, \quad x \in \text{MR}.$
4.2 Descriptors of Local Appearance

When considering a pair of patches for a correspondence, these twelve parameters are combined to provide local transformations (both geometric and photometric) between the images. The transformations are exploited later during the matching step to reject improbable and inconsistent matches as described in Section 4.3.

Figure 4.1 illustrates the normalisation procedure. On query (a) and database (f) images, MSERs are detected and LAFs constructed, independently on each image. Geometric normalisation according to the transformation between detected LAFs and the canonical coordinate system yields patches depicted in columns (b) and (e). The patches are then normalised photometrically, as shown in columns (c) and (d). These geometrically and photometrically normalised patches represent local object appearance.

4.2 Descriptors of Local Appearance

A descriptor is a suitable representation of a local image patch. It is associated with a similarity measure, often Euclidean distance. Because of the normalisation, any representation of the normalised patches (shown in Figure 4.1 (c) and (d)) is invariant to affine geometric and diagonal photometric transformations. There is therefore no need for e.g. rotation invariance of the representation. Obviously, directly the intensities of the normalised regions can be stored, and a pixel-wise difference used as the similarity measure, but such a representation is sensitive to image noise and to imprecise normalisation.

In the following we summarise what we found that are desirable properties of a descriptor. A descriptor has to be discriminative, i.e. to be able to distinguish between a large number of image regions. The value of the similarity measure should well separate corresponding and not-corresponding regions. The descriptor should be insensitive to localisation errors of the detector, i.e. to misalignment of corresponding patches, and invariant to image transformations not covered by the detector covariance. If the detector, for example, does not resolve rotation (as various feature point detectors do not) rotational invariants have to be used as descriptors. In our case, local affine frames provide covariance with affine transformations of the image. The descriptor should thus be insensitive to small perspective distortion and to distortions caused by non-planarity of the surfaces. Finally, the descriptor should be efficient from the computational point of view. The data representation should be compact, memory efficient, and fast to construct. More importantly, efficient evaluation of similarity of two descriptors is required.

Discrete Cosine Transformation We propose to represent the local appearance by low-
Figure 4.2: Examples of correspondences established between frames of query (left columns) and database (right columns), for the image pair from Figure 3.14. (a) geometrically and photometrically normalised image patches, (b) the same patches reconstructed from 10 DCT coefficients per colour channel.

frequency coefficients of the discrete cosine transformation (DCT). For uniformly distributed data, the DCT approximates [Jai86] the Karhunen-Loeve transformation (KLT), which is widely used in pattern recognition to reduce data dimensionality without significant deterioration of recognition rate. DCT has the desirable properties of a descriptor. It is computationally efficient, fast algorithms exist that computes DCT with $O(n \log n)$ time complexity. Hardware implementations of DCT are widely available due to its widespread use in image and video compression (JPEG, MPEG, etc.). The definition of two-dimensional DCT for an input normalised patch $\hat{I}$ and output matrix of coefficients $D$ is:

$$D_{p,q} = \alpha_p \alpha_q \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \hat{I}_{m,n} \cos \frac{\pi (2m+1)p}{2N} \cos \frac{\pi (2n+1)q}{2N},$$  

where $N$ is the patch resolution in pixels, $0 \leq p \leq N$ and $0 \leq q \leq N$ are coefficient indices, and

$$\alpha_p = \begin{cases} 1/\sqrt{N} & \text{if } p = 0 \\ \sqrt{2/N} & \text{if } 1 \leq p \leq N - 1 \end{cases}, \quad \alpha_q = \begin{cases} 1/\sqrt{N} & \text{if } q = 0 \\ \sqrt{2/N} & \text{if } 1 \leq q \leq N - 1 \end{cases}.$$

Robustness to frame misalignment is achieved by storing only low-frequency coefficients, which are less sensitive to the misalignment than higher frequencies. Discriminativity of the DCT representation depends on the number of coefficients stored. In Section 6.2.9, it is experimentally shown how the number of coefficients affect the recognition performance. In another experiment, in Section 6.2.8, we demonstrate that the DCT representation outperforms descriptor composed of directly the normalised pixels and that it has about the same discriminative potential as the widely used SIFT descriptor [Low04]. Figure 4.2 (b) shows an example of patches reconstructed from 10 DCT coefficients. The image patches are the same as in Figure 4.2 (a).
4.3 Matching: Forming Tentative Correspondences of Local Regions

Let us have a set \( S^D \) of frames \( F^D \) detected on a single database image, and a set \( S^Q \) of frames \( F^Q \) detected on a query image. Let each frame be associated with a descriptor of normalised local appearance. The set of tentative correspondences \( T \) is a subset of \( S^D \times S^Q \) where \( \times \) denotes the cartesian product. A frame pair \( \{F^D, F^Q\} \in T \) if \( F^D \) and \( F^Q \) have similar descriptors, i.e. if the frames are tentatively corresponding on the basis of local measurements. The correspondences in \( T \) include many outliers as they are based solely on the properties of the two frames in question, regardless of other correspondences on the objects. At a later stage, the correspondences are verified and pruned according to consistency with a global model. Different strategies are employed to obtain the set \( T \):

**Nearest match** This is the most commonly used strategy, used in all the experiments in Chapter 6: For query frame \( F^Q \in S^Q \) find its closest database frame \( F^D \in S^D \): \( F^D = \text{argmin}_i(d(F^Q, S^D_i)) \). Then \( \{F^Q, F^D\} \in T \) iff \( d(F^Q, F^D) < \Theta_d \), where \( d \) is a “distance” function (discussed later), and \( \Theta_d \) is a threshold on \( d \).

**Mutually nearest match** This strategy is suitable for symmetric matching problems, e.g. for wide-baseline stereo matching. The fraction of correct correspondences (inliers) in \( T \) is increased, causing the successive global consistency check execute faster. But the absolute number of inliers is typically reduced. For query frame \( F^Q \in S^Q \) find its closest database frame \( F^D \in S^D \): \( F^D = \text{argmin}_i(d(F^Q, S^D_i)) \). Then \( \{F^Q, F^D\} \in T \) iff \( d(F^Q, F^D) < \Theta_d \).

**All (or \( \forall \) most) similar** This strategy is used when repetitive structures are expected on the objects of interest. Repetitive structures induce ambiguous correspondences, which cannot be resolved at the time of forming of \( T \). Here, each query frame is associated with a set of possibly corresponding frames – of which at most one is correct. The decision about which of the correspondences is the correct one (if any) is left to the phase of verification of the global consistency. The drawback is in higher number of false correspondences (outliers), leading to increase of the computational load of the consistency check, or even to its failure due to small fraction of inliers: For each frame \( F^Q \in S^Q \) find all near frames (or at most \( \forall \) closest frames) \( F^D \in S^D \). \( \{F^Q, S^D_i\} \in T \) iff \( d(F^Q, S^D_i) < \Theta_d \).

The distance function \( d \) is a scalar function expressing similarity of two frames. Besides reflecting the similarity of the descriptors of the normalised patches, it might include terms related to the probability of the geometric and photometric transformations between the two frames. Let \( F^D \) and \( F^Q \) denote the frames on query resp. database images. Let \( A^D \) and \( A^Q \) be the affine geometric transformations which transform the canonical coordinate system into image coordinates of the respective frames (as in Section 3.4.1). Finally, let \( P^D \) and \( P^Q \) be the photometric transformations of the RGB values transforming the normalised intensities to the corresponding intensities in the images. Then the transformations \( A^{QD} = A^D(A^Q)^{-1} \) and \( P^{QD} = P^D(P^Q)^{-1} \) are the geometric resp. photometric transformations between the images – if the frames \( F^D \) and \( F^Q \) correspond. The situation is illustrated in Figure 4.3.

Although allowing arbitrary transformations \( A^{QD} \) and \( P^{QD} \) accomplishes geometric and photometric invariance, some transformations are unlikely to happen in practice. The probability distributions of \( A^{QD} \) and \( P^{QD} \) are therefore estimated from training scenes, and the frame distance \( d \) is penalised for unlikely transformations. In our experiments the distributions are approximated by step functions. If the transformations are out of certain, problem-specific limits, the frame pair will not match, i.e. \( d \) evaluates to infinity. If they are within the limits, no
penalty is imposed, and \( d \) evaluates directly to the similarity of the descriptors. It allows the function \( d \) to be implemented as a fast sequence of thresholdings, with the descriptor similarity evaluation at the end of the sequence.

The geometric transformation \( A^{QD} \) is decomposed into three values. Scale:

\[
s = \sqrt{|A^{QD}|},
\]

anisotropy:

\[
a = \sqrt{\frac{\max(e_1,e_2)}{\min(e_1,e_2)}},
\]

where \( e_1 \) and \( e_2 \) are the eigenvalues of \( \overline{AA}^T \): \( e_1,e_2 = \text{eig}(\overline{AA}^T) \) and \( \overline{A} \) is the linear \( 2 \times 2 \) submatrix of \( A^{QD} \), and finally rotation of a vertical direction:

\[
r = \tan^{-1}(\overline{A} \cdot (0,1)^T).
\]

The frames \( F^Q \) and \( F^D \) are not matched (i.e. \( d \) evaluates to infinity) whenever

\[
s > \Theta_{\text{MAXSCALE}} \quad \text{or} \quad s < \Theta_{\text{MINSCALE}} \quad \text{or} \quad a > \Theta_{\text{ANISSCALE}} \quad \text{or} \quad |r| > \Theta_{\text{ROTATION}}.
\]

The photometric transformation \( P^{QD} \) is parameterised by multiplicative factors of individual R,G,B channels and by offset of chromaticity of average colour of the patches associated to the frames. Let \( \mu_i^Q, \mu_i^D, \sigma_i^Q \) and \( \sigma_i^D, \ i \in \{R,G,B\} \), denote the mean values, resp. variances of values in individual R, G, B channels of the query and database patches respectively. These values are obtained during the photometric normalisation. Finally, let \( c^Q \), resp. \( c^D \) denote the average patch colour in chromatic plane, i.e.

\[
c^Q = \left( \frac{\mu_R^Q}{\mu_R^Q + \mu_G^Q + \mu_B^Q}, \frac{\mu_G^Q}{\mu_R^Q + \mu_G^Q + \mu_B^Q}, \frac{\mu_B^Q}{\mu_R^Q + \mu_G^Q + \mu_B^Q} \right), \quad c^D = \left( \frac{\mu_R^D}{\mu_R^D + \mu_G^D + \mu_B^D}, \frac{\mu_G^D}{\mu_R^D + \mu_G^D + \mu_B^D}, \frac{\mu_B^D}{\mu_R^D + \mu_G^D + \mu_B^D} \right)
\]

The frames \( F^Q \) and \( F^D \) are not matched if there is a high change in contrast, i.e.

\[
\max \left( \frac{\sigma_i^Q}{\sigma_i^D}, \frac{\sigma_i^D}{\sigma_i^Q} \right) > \Theta_{\text{CONTRAST}} \quad i \in \{R,G,B\}
\]

or significant difference in chromaticity, i.e.

\[
||c^Q - c^D|| > \Theta_{\text{CHROMATICITY}}.
\]
4.4 Globally Consistent Subset of Tentative Correspondences

If the decomposed transformations are within all the thresholds, \(d\) evaluates to the similarity of the descriptors of local appearance. The thresholds \(\Theta_{\text{MAXSCALE}}, \Theta_{\text{MINSCALE}}, \Theta_{\text{ANISSCALE}}, \Theta_{\text{ROTATION}}, \Theta_{\text{CONTRAST}},\) and \(\Theta_{\text{CHROMATICITY}}\) differ for specific recognition problems. For example, the threshold \(\Theta_{\text{ROTATION}}\) applies only in tasks for which it is known that the mutual orientation between the camera and the objects is fixed. An example might be a camera mounted in upward orientation, and the orientation of objects of interest, for example buildings, being inherently fixed. Or, if the objects are taken from constant distance, as in the COIL-100 dataset (see Section 6.1.2), the scale thresholds are set to small values, i.e. \(\Theta_{\text{MAXSCALE}} = 2^2\) and \(\Theta_{\text{MINSCALE}} = 2^{-2}\). For other datasets, e.g. the LAFsTest dataset (Section 6.1.5), where large scale changes occur, we set \(\Theta_{\text{MAXSCALE}} = 10^2\) and \(\Theta_{\text{MINSCALE}} = 10^{-2}\). The thresholds are estimated from training instances by inspection of histograms of transformations of formed correspondences, and represent a trade-off between invariance and limited number of false matches.

4.4 Globally Consistent Subset of Tentative Correspondences

The process of obtaining tentative correspondences by pair-wise matching of local frames and their descriptors does not take into account the mutual relation between frames. It might for example happen that one of the tentative correspondences implies that the object is larger in the query image than in the model image, while another correspondence suggests that it is smaller and perhaps rotated. Such correspondences, although perfectly possible on their own, are not mutually consistent (assuming the object is rigid). A subset of the obtained tentative correspondences is therefore sought where all correspondences are consistent with some global object model.

The first issue is the choice of the type of the global model. For rigid 3D objects the correct model is a 3D model imposed through epipolar geometry. A method for estimating epipolar geometry from LAF correspondences is described in [CMO04]. The method takes advantage of the fact that a frame correspondence provides an affine transformation between the images, and consequently only three correspondences suffice to obtain the epipolar geometry. For planar objects, the global model is a homography. For deformable non-rigid (but not articulated) objects, iterative method described by Ferrari in [FTG04] can be used, although it is rather slow for practical exploitation.

For the purpose of object recognition, we use planar models. Unless we are recognising whole complex scenes (e.g. interior of a building), the depth of the visible part of an objects is typically too shallow to allow for reliable epipolar geometry estimation. We found it sufficient to model the object either as a single planar surface, or as a set of planar surfaces.

Let us have two tentative correspondences, between frames \(F^Q_1\) and \(F^D_1\), and between \(F^Q_2\) and \(F^D_2\) respectively. Each correspondence suggest geometric \(A_{1QD}^1\) resp. \(A_{2QD}^1\) and photometric \(P_{1QD}^2\) resp. \(P_{2QD}^2\) transformation between the images (see Figure 4.3). Would the frames lie on the same planar surface, the geometric transformations would be identical up to a perspective distortion and an imprecision in frame localisation. Assuming light sources at infinity and no shadows nor specular reflections across the planar surface, the two photometric transformations would also be identical.

The set of tentative correspondences \(T\) is decomposed to subsets of consistent correspondences, i.e. subsets in which all correspondences imply identical (up to a small tolerance) image-to-image transformation. Each subset represents a single 3D plane in the scene. Subsets of low cardinalities (typically lower than 4) are rejected as outliers, and the decision about the presence of an object in the scene relies only on the correspondences in subsets of high cardinality. In our experiments, we rank the database objects by the number of consistent correspondences, weighted by the descriptor similarity \(d\).
5 Tentative Correspondences using Decision Trees

Realistic approaches to recognition, detection and localisation of objects from large collections must support sub-linear indexing, i.e. the ability to associate query input with objects represented in the memory, at a speed that does not significantly depend on the number of images and objects remembered. Any technique that compares the query one-by-one with stored models is linear in the number of known objects. Such recognition techniques, solving effectively a sequence of two-image matching problems, will have, sooner or later, an unacceptable response time.

5.1 Introduction to the Decision Trees

Searching and indexing are well-studied subjects, and two sub-linear methods dominate the field – hashing and tree search. This chapter presents an approach that achieves sub-linear, real-time recall by representing the visual memory as a binary decision tree, organised to minimise average time to decision.

Besides efficient retrieval, the decision tree approach allows to depart from the “compute a fixed-size feature vector on a fixed measurement region” paradigm. In this paradigm, followed by all approaches reviewed in Section 2.3, the local reference frame is described by a function of pixel values from a measurement region whose size and shape, if expressed in the local frame coordinates, is fixed. Also in Chapter 4, the shape was a square of a predefined size. It is clear that a fixed measurement region will lead to difficulties when recognising certain classes of objects, e.g. “wire-like” objects as bicycles, where any square neighbourhood includes background. Perhaps more significantly, a measurement region of a certain size will be too big for some frames, e.g. including parts of background or discontinuities, and yet it will be too small for other frames whose descriptors will not be discriminative.

The problem of a better-than-fixed measurement region seems insurmountable. How can possibly be the measurement region adapted unless we know what we are looking at? We finesse the problem by interleaving the processes of recognising the frame and deciding where to measure next. Which measurements are used depends on the particular frame, and measurements outside the objects (on scene background) are avoided. The frame is recognised by descending a decision-measurement tree, where each decision not only reduces the number of potential corresponding frames represented in the tree, but also defines which measurements are taken next. More precisely, a binary tree is formed in the learning stage. For each non-terminal node, a binary valued measurement-decision function, called a ‘weak classifier’, is selected from a large pool according to an optimisation criterion. The criterion is a lower bound on the expected time to decision. The term ‘weak classifier’ stresses analogies with discrete AdaBoost - a classifier is selected by a greedy algorithm, it could be any binary function of pixel values and, as will be shown later, it is not required to make unequivocal decisions.

Establishing tentative correspondences with the decision-measurement tree has a number of favourable properties. The advantages of a data-specific measurement region have already been mentioned. From a computational point of view, efficiency of recognition is increased since only a small fraction of potential measurements is evaluated. In case that a measurement is close to a decision boundary, or not available at all as in the case when it is taken from the background, robustness of the search is easily achieved by inserting the learned frame in both subtrees. With this modification, the search in the recognition stage descend always into only a single branch,
Figure 5.1: Formation of tentative correspondences. (a) Sequential matching, each query frame is compared against every database frame. (b) Logarithmic-time matching using a decision tree.
guaranteeing that a leaf of the tree is reached in $\log(N)$ steps, where $N$ is the number of frame instances stored in the tree. Last but not least, the learning process explicitly takes into account geometric uncertainty and image statistics to minimise the response time (see Section 5.2).

Decision trees were successfully applied to various recognition and classification problems, see e.g. [Mur98] for a survey. Our problem differs from the bulk of published work in not having the data labelled. Unsupervised learning techniques are exploited, the tree provides an automatic clustering of image patches by their visual similarity. Our work on decision trees was inspired by Lepetit and Fua [LLF05], whose approach, however, differs from ours in several areas. First, they set the tree size (and the number of trees, since they are using multiple randomised trees) by hand, while in our approach the tree size is a function of image database content. Second, our measurements are invariant to affine deformations of the image (due to the LAF constructions), thus a 3D model, or synthetically warped 2D images capturing the appearance variations, are not needed. We also explicitly consider image noise and background segmentation of the measurements, while Lepetit et al. synthetically generate noisy patches and patches with random background.

### 5.2 Recognition with Decision-Measurement Trees

This section describes the decision-measurement tree which is used to represent the “visual memory”. A decision tree is a tree structure where a simple test (a weak classifier) is assigned to each non-terminal node. Each leaf corresponds to a volume of observation space, that is bounded by the sequence of decisions made on the path from the tree root to that particular leaf. During the recall phase, the tree is traversed according to the decisions at non-terminal nodes, until a leaf node is reached. The elements in the leaf do not necessarily match the query – being in the same volume does not imply proximity – and an additional evaluation of a similarity measure is necessary to distinguish matching and non-matching elements. Recall can be viewed as a sequential reduction of the set of candidate correspondences until a subset of a small predefined cardinality (called “leaf capacity”) is reached. The elements remaining in the subset are sequentially searched for matches.

In the experiments in Section 6.3 we employ only one simple type of weak classifiers, but multiple types can be freely combined within the tree. Our classifiers are binary functions $d_{\mathbf{x}, \Theta_{\mathbf{x}}}: F \to \{L, R\}$, which threshold a single pixel value. Vector $\mathbf{x}$ is specifying the measurement location, $\Theta_{\mathbf{x}}$ is a scalar threshold on the value at $\mathbf{x}$, $F$ is a local affine frame, and $\{L, R\}$ are the decisions to search left and right subtrees respectively.

#### 5.2.1 Retrieval

Due to noise, the decisions are ambiguous for values close to the thresholds $\Theta_{\mathbf{x}}$. The ambiguity can be solved in the recognition phase by descending both subtrees, as e.g. in the classical kD-tree algorithm. In an alternative approach, the elements are in ambiguous cases stored redundantly in both subtrees, as e.g. in spill-trees [Liu06]. There is then no need to backtrack or split the tree search during recognition (recall), all uncertainties are addressed in the training phase. This approach allows for faster retrieval at the expense of memory needed for the redundant representation. Since our motivation is to achieve high recognition speeds, we have adopted the second approach. The design leads to a straightforward retrieval algorithm (see Algorithm 4). The retrieval is very fast, since for each query frame $F$ only one evaluation of a weak classifier (thresholding of a single pixel value) is performed at each tree level. The depth of the tree is typically 15 to 25, depending on the database size.
Algorithm 4 Decision tree: Retrieving stored frames

**Input:** $F$: a query local affine frame  
**Output:** $S$: a set of candidate matches

function $\text{Tree}.\text{retrieveFrames}(F) \to S$

$S := \text{root}.\text{retrieveFrames}(F)$

function $\text{Node}.\text{retrieveFrames}(F) \to S$

if isLeaf then  
$S := \{F_i : F_i \in \text{leafFrames} \land d(F, F_i) < \Theta_d\}$ /*distance function $d$ described in Section 4.3*/

else  
if $d_{x,\Theta_x}(F) = L$ then  
$S := \text{leftSubtree}.\text{retrieveFrames}(F)$

else /*$d_{x,\Theta_x}(F) = R$*/  
$S := \text{rightSubtree}.\text{retrieveFrames}(F)$

5.2.2 Learning the Tree

A separate tree is constructed for every type of LAF construction. Starting with a set $S_F$ of frames of a single type, collected from all database images, the set is recursively divided into subsets at non-terminal nodes. Non-terminal nodes are inserted until (a) the cardinality of the particular subset is below a predefined threshold, the 'leaf capacity', or (b) the frames in the subset are mutually matching, i.e. the distance function $d$ is under the threshold $\Theta_d$ (described in Section 4.3) for each pair of frames in the subset. The condition (b) accommodates for the situation where there are multiple images of the same object in the database, or when the objects contain repetitive structures.

Let $r(F)$ denote a random realisation of frame $F$ in a query image. The random function $r$ encapsulates geometric and photometric misalignments between corresponding frames, as well as image noise, blur and other image distortions. Algorithm 5 ensures that $F$ is represented in every leaf where the probability of a query realisation $r(F)$ falling to that leaf is above a threshold $\Theta_p$; $\Theta_p$ is a parameter of the method. The probability that a query realisation $r(F)$ of frame $F$ will descend the left $(p(d_{x,\Theta_x}(r(F)) = L))$, and right $(p(d_{x,\Theta_x}(r(F)) = R))$ subtree respectively, given a classifier $d_{x,\Theta_x}$ and the frame $F$ is analysed below.

Estimation of geometric precision of the MSER-LAF method. Local affine frames were constructed on several image pairs related by known homographies. Corresponding frames do not align perfectly – a single spot in the scene occurs at slightly different pixel positions. Figure 5.2 shows covariance matrices of distributions of pixel displacements, estimated on thousands of frames. The distributions represent a localisation uncertainty $l_x$ of pixels in query frames, $x$ is the pixel location in the normalised frame. As expected, the farther from the detected frame, the larger is the uncertainty. It is clear that the distributions differ significantly for different types of frame constructions. A separate set of distributions is therefore maintained for each frame type.

Considering the estimated geometric uncertainty. Given a database frame $F$ of a certain type, what is the probability of observing value $v$ at measurement position $x$ in a corresponding query frame $r(F)$? The situation is depicted in Figure 5.3. Figure 5.3 (a) illustrates a part of the frame neighbourhood around measurement position $x$, and the corresponding distribution of localisation uncertainty $l_x$ for that particular frame type. The probability $p(v)$ of
Algorithm 5 Decision tree: Learning

**Input:** \( S_F \): Set of LAFs of one type

**procedure Tree.build \((S_F)\)**

\[ S := \emptyset \]

for all \( F \in S_F \) do

\[ S := S \cup \{(F, 1)\} \] /*assign unit probability*/

root.build \((S)\)

**procedure Node.build \((S)\)**

if \(|S| \leq \text{leaf capacity} \) or indistinguishable \((S)\) then

\[ \text{isLeaf} := \text{true}, \text{leafFrames} := S \]

else

\[ d_{\mathbf{x}, F} := \text{selectClassifier} \((S)\) \]

\[ S_L = \emptyset, S_R = \emptyset \]

for all \( \{F, p_F\} \in S \) do

\[ p_L := p_F \cdot p(d_{\mathbf{x}, F}(r(F)) = \text{L}) \]

\[ p_R := p_F \cdot p(d_{\mathbf{x}, F}(r(F)) = \text{R}) \]

if \( p_L \geq \Theta_p \) then

\[ S_L := S_L \cup \{(F, p_L)\} \]

if \( p_R \geq \Theta_p \) then

\[ S_R := S_R \cup \{(F, p_R)\} \]

if \( S_L \neq \emptyset \) then

leftSubtree.build \((S_L)\)

if \( S_R \neq \emptyset \) then

rightSubtree.build \((S_R)\)

observing a value \( v \) in a query frame \( r(F) \) at position \( \mathbf{x} \) is given as

\[
p_{\mathbf{x}, F}(v) = \int_{\Omega_{v,F}} l_{\mathbf{x}} \, d\Omega,
\]

where \( \Omega_{v,F} \) is the area in \( F \) covered by pixels of value \( v \). Figure 5.3 (b) shows the resulting distribution \( p_{\mathbf{x}, F}(v) \) for the example from Figure 5.3 (a). Narrow distributions of \( p_{\mathbf{x}, F}(v) \), which are benign for unambiguous decisions about query frames, are intuitively obtained either in areas of uniform intensity (\( v \) is constant over \( l_{\mathbf{x}} \)), or where the localisation is precise – \( l_{\mathbf{x}} \) “covers” only a few pixels, ideally a single one.

The framework also consistently handles situations when some of the measurements are undefined, e.g. because not being on the object. Imagine hand-segmented model images where the outline of the object is available (as in Figure 5.4 (a)). Some of the frames will partially cover an area not on the object. In this area, the model cannot predict what value \( v \) will occur in a query frame. Without a background model (the probability distribution of intensities in the scene background), all values \( v \) are considered equiprobable. That is, if \( \mathbf{x} \) is known to be outside of the object, \( p_{\mathbf{x}, F}(v) \) has flat distribution over the whole domain of \( v \) \((p_{\mathbf{x}, F}(v) = 1/256 \) for \( v \in \{0, \ldots, 255\}\)).

**Modelling photometric noise.** A very simple model of photometric noise is employed – the noise distribution is assumed to be flat in a range of \((-\epsilon, \epsilon)\) intensity values. As illustrated in Figure 5.3 (c), the probability of observing value \( v \) becomes \( p_{\mathbf{x}, F}(v)/2\epsilon \) over the \( \epsilon \)-range. In the experiments, \( \epsilon \) is set to 10, independently of \( v \).
5 Tentative Correspondences using Decision Trees

Figure 5.2: Geometric misalignment of detected frames, experimentally obtained for different types of frame constructions. The images show covariance matrices of distributions of displacements of pixels in normalised neighbourhoods of detected LAFs. (a) LAF construction based on normalisation by region covariance matrix (Figure 3.13(a)), (b) LAF construction based on a bi-tangent segment (Figure 3.13(d)), (c) LAF construction based on normalisation by covariance matrix of a concavity (Figure 3.13(i))

Figure 5.3: Probability of observing value $v$ at position $\mathbf{x}$ in a query realisation of frame $F$. (a) estimated localisation uncertainty $l_x$ for a pixel at position $\mathbf{x}$, (b) probability $p_{x,F}(v)$ of observing value $v$ at $\mathbf{x}$ in $r(F)$, (c) the probability after considering photometric noise

Going back to Algorithm 5, the probabilities that a query realisation $r(F)$ will descend into the left and right subtree respectively are expressed as

$$p(d_{x,\Theta_x}(r(F)) = L) = \int_{0}^{\Theta_x} p_{x,F}(v) \, dv, \quad \text{resp. } p(d_{x,\Theta_x}(r(F)) = R) = \int_{\Theta_x}^{255} p_{x,F}(v) \, dv \quad (5.2)$$

for $v \in \{0 \ldots 255\}$ thresholded by $\Theta_x$.

The remaining issue in the tree construction algorithm is the choice of weak classifiers for non-terminal nodes. The objective is to minimise the expected recall time for query frames. To select the classifier, let us have a set $\mathcal{S}$ of frames $F$, each with assigned probability $p_F$ – the probability that $r(F)$ will descend from root to that node. The task is to select a measurement position $\mathbf{x}$ and a threshold $\Theta_{x}$ so that, on average, the queries reach leaf nodes in minimal time, i.e. on minimal tree level. The requirements translate to (a) that the tree is balanced for query frames and (b) the number of ambiguous frames stored in both subtrees is minimised. It follows
Concluding Remarks

Figure 5.4: The need for variable-sized measurement regions. (a) An example of a segmented model image and some of its normalised patches. Using a common fixed measurement region where values are defined for all frames would lead to small non-discriminative descriptors. Large regions would include background in query images. (b) Frames detected on multiple instances of the 'e' letter on the 'Multiple view geometry' book title. The instances cannot be distinguished close to the detected frames and a distant measurement (e.g. on a neighbouring letter) is needed to separate them.

from (a) that for any particular \( \bar{x} \), the threshold \( \Theta_{\bar{x}} \) is set to median value, so that

\[
\sum_{F \in \mathcal{S}} p_F p\left(d_{\bar{x}, \Theta_{\bar{x}}}(r(F)) = L\right) = \sum_{F \in \mathcal{S}} p_F p\left(d_{\bar{x}, \Theta_{\bar{x}}}(r(F)) = R \right), \text{ i.e.}
\]

\[
\sum_{F \in \mathcal{S}} p_F \int_{0}^{\Theta_{\bar{x}}} p_{\bar{x}, F}(v) \, dv = \sum_{F \in \mathcal{S}} p_F \int_{\Theta_{\bar{x}}}^{255} p_{\bar{x}, F}(v) \, dv \quad (5.3)
\]

The measurement position \( \bar{x} \) that best separates (minimises ambiguity) of the frames in \( \mathcal{S} \) is selected as

\[
\bar{x} = \arg\min_{\bar{x}} \sum_{F \in \mathcal{S}} \min \left( p_F p\left(d_{\bar{x}, \Theta_{\bar{x}}}(r(F)) = L\right), p_F p\left(d_{\bar{x}, \Theta_{\bar{x}}}(r(F)) = R \right) \right), \quad (5.4)
\]

with \( \Theta_{\bar{x}} \) given by Eq. 5.3. Ideally, when a position \( \bar{x} \) (and a corresponding threshold \( \Theta_{\bar{x}} \)) is found which perfectly separates the set \( \mathcal{S} \), the minimised term evaluates to zero. In the worst case of identical distributions \( p\left(d_{\bar{x}, \Theta_{\bar{x}}}(r(F)) \right) \) for all \( F \in \mathcal{S} \), the term evaluates to 0.5 (after normalisation by \( \frac{1}{|\mathcal{S}|} \)). Let us consider the example shown in Figure 5.4 (b). No measurement positions \( \bar{x} \) on the letter “e” nor the brown background will allow for discrimination of the frames. Due to formula 5.4, a distant but discriminative measurement is rather selected.

The learning process takes several hours. A learned tree is determined by node classifiers and by identification of database frames contained in leaves. It is not necessary to relearn the tree every time an image is added or removed from the database, but additional images lead to increasingly sub-optimal representation, as it is no longer guaranteed that each leaf contains at most “leaf capacity” of frames.

5.3 Concluding Remarks

A method capable of sub-linear formation of tentative correspondences, and subsequently of sub-linear recall of database images, is proposed. The frames are stored in a binary decision-measurement tree, which is organised to minimise average time to decision. A query frame is recognised by descending the tree where each decision not only reduces the number of potential corresponding frames, but also defines which measurements are taken next. When a leaf node is
reached, a limited number of frame-to-frame similarity evaluations is performed to obtain tentative frame correspondences. Robustness of the correspondence search is achieved by explicitly accounting for the geometric localisation uncertainty of the MSER-LAF detector.

The performance of the proposed method, both in the recognition rate and execution speed, is evaluated in Chapter 6. There we show, that the method supports near real-time recognition of hundreds of real-world objects with state-of-the-art recognition rates. Establishing correspondences between hundreds of query local frames and hundreds of thousands of stored frames takes only a few milliseconds.
6 Experimental Validation

In this chapter the performance of the proposed object recognition system is evaluated. First, Section 6.1 introduces the datasets which are used in the experiments in the rest of the chapter. Section 6.2 then evaluates several aspects of the recognition system: repeatability of MSER and LAF detectors is measured, and impact on recognition performance of various implementation decisions, such as selection of data representation or of various thresholds, is assessed. In Section 6.3 results on publicly available large-scale datasets are presented. Achieved recognition results are compared to other published results. In the last section, the computational complexity of the system is analysed and bottlenecks are identified.

6.1 Datasets

This section describes various datasets, which were used for experimental validation thorough this chapter.

6.1.1 Mikolajczyk’s Dataset

In [MTS+05], Krystian Mikolajczyk et al. studied repeatability of various affine invariant detectors. For this purpose, a database of images was created. The images depict planar objects, thus homographies, which are known for the data, describe geometric transformations. Localisation of detected primitives can be therefore directly compared across different views. We use a subset of this dataset (shown in Figure 6.1) to evaluate repeatability and localisation precision of MSER detector (described in Section 3.2.1) and of LAF construction processes (described in Section 3.4).

Figure 6.1: Images used to evaluate the MSER/LAF repeatability. The homography between the images is known. (a) a reference image, which is matched to remaining images (b) of each set.
6 Experimental Validation

![Image of COIL-100](image1.png)

Figure 6.2: COIL-100: Examples of objects from the database.

![Image of ZuBuD](image2.png)

Figure 6.3: ZuBuD dataset [SSG03]: Examples of (a) query and (b) corresponding database images.

6.1.2 COIL-100 Dataset

The Columbia object image library (COIL-100)\(^1\) dataset has been widely used in object recognition literature [VHM04, LS02, CHPN02, YRA00], and serves therefore as a benchmark. The database consists of color images of 100 different objects. 72 images were acquired of each object, at pose intervals of 5°. The images were preprocessed so that either the object’s width or height (whatever is larger) fits the image size of 128 pixels. Several objects from the database are shown in Figure 6.2. The images are captured in a controlled environment, neither occlusion, background clutter, nor illumination changes are present. It makes the dataset suitable especially for pure appearance-based methods, which represent whole images as a single entity. A dense sampling of the training poses is typically used, taking every second image (10° apart) for training. The query images are then only 5° distant from closest database image.

6.1.3 ZuBuD Dataset

The ZuBuD dataset represents a larger, real-world problem, with images taken outdoor, with occluded objects, varying background, and illumination changes. ZuBuD contains images of 201 buildings in Zurich, Switzerland, and is publicly available [SSG03]. The database consists of five photographs of each of the 201 buildings, 1005 images in total. Image resolution is 320 × 240 pixels. The photographs are taken from different viewpoints but under approximately constant illumination conditions. A separate set of 115 query images is provided. For every query image, there are exactly five matching images of the same building in the database. Not all the database buildings have corresponding queries, the number of queries per building ranges from 0 to 5. Query and database images differ in viewpoint; variations in illumination are present, but rare. Examples of corresponding query and database images are shown in Figure 6.3.

\(^1\)http://www.cs.columbia.edu/CAVE

64
6.1.4 Focus Dataset

The FOCUS dataset represents a typical retrieval problem, product logos are sought in scanned advertisements. The database contains 360 colour high-resolution images of advertisements scanned from miscellaneous magazines. A set of query logo images is provided, the logos typically occupy only a small portion, e.g. 1%, of the commercial. Example queries and commercials identified from the database are shown in Figure 6.4. There is little variability in the logo appearance, the only geometric transformation present is scaling. What makes this dataset challenging is the amount of structured background, which increases the risk of false recognition response.

6.1.5 LAFsTest Dataset

The set contains 15 database images of household objects and 170 queries containing one or more of the objects. Images were captured by a DV camcorder and downsized by a factor of two to \(360 \times 288\) pixels. Objects have a well structured surface albedo, as is suitable for our recognition system. There are however significant geometric variations, the objects are scaled up to about \(10\times\), and the viewing angle differs up to \(80^\circ\) from the database view. Database and query images were captured at different time, so illumination changes, if not severe, are present. Figure 6.5 shows examples of images from this dataset.

6.2 Evaluation of Components of the Recognition System

This section first analyses repeatability and localisation precision of MSER region detector and of LAF construction. The analysis is done on Mikolajczyk’s dataset, for which location of corresponding image regions is known. In following sections several aspects of the system are examined where multiple implementation variants were considered, or where a certain parameter, as e.g. a threshold, was applied. The obtained results allows for an informed choice about appropriate configuration of the method.
Experimental Validation

6.2.1 MSER Repeatability

An evaluation of repeatability of regions detected by the MSER detector (Section 3.2.1) is given here. Localisation precision and percentual repeatability is analysed on three scenes from Mikolajczyk’s dataset (Section 6.1.1). The repeatability is compared across various ordering of colour pixels (described in Section 3.2.2). For evaluation on more comprehensive datasets, and for comparison with other detectors, refer to [MTS+05].

The following protocol was used. (i) MSER regions are detected in all images, (ii) regions from the reference image from each set (Figure 6.1 (a)) are projected by the ground-truth homographies onto the remaining images (Fig 6.1 (b)), and (iii), for each projected MSER region from the reference image a single region with best alignment \( a \) (see later) is selected on each of the remaining images.

We want to investigate how many of the detected regions are repeated, and how well are the repeated regions aligned. Figure 6.6 illustrates how we approximate the region alignment \( a \). Fig. 6.6 (a) shows an example of two corresponding regions, one computed in current image, second one projected from the reference image by the ground-truth homography. The alignment is computed as the ratio of the area of intersection (dark region in Fig. 6.6 (b)) and the area of union of the regions, i.e. the alignment \( a = \frac{|\Omega'_1 \cup \Omega_2|}{|\Omega'_1 \cap \Omega_2|} \), where \( \Omega'_1 \) is the region projected from the reference image, \( \Omega_2 \) is a detected region, and \( |\ldots| \) denotes the region area. Our results cannot be directly compared to the results presented in [MTS+05], since there the regions are first approximated by ellipses.

The results are shown in Figures 6.7, 6.8, and 6.9. In these figures, (a)s show the reference images, which were matched against the remaining images of each set. The other charts display observed region alignments in the form of cumulative histograms. So, for example, the reading at 0.7 horizontal tick says how many of the regions were found with alignment \( a \) being at least 0.7. Or, would we define a successfully repeated detection as having the alignment \( a \) at least 0.7, how many of the regions were repeated. Higher placed curves indicate better repeatability and alignment. The MSERs are evaluated with respect to two parameters: the pixel ordering (see Section 3.2.2) used to obtain the regions, and the region “stability”, which is given as the number of consecutive thresholds over which the region does not change (see Section 3.2.1).

Figures 6.7, 6.8, and 6.9 (b) show region alignment obtained for different viewpoints, i.e. for different images from the respective sets. Regions of all stabilities and for all pixel orderings
6.2 Evaluation of Components of the Recognition System

Figure 6.6: Computation of region alignment. (a) An example of two corresponding detected regions. (b) Intersection (dark area) and union (dark and bright area combined) of the regions.

are combined here. The order of the curves corresponds to the order of images in Figure 6.1, only the “graffiti” set is labelled with viewing angle differences. Figures in second rows, (c) and (d), show the alignments observed on regions with different orderings of colour pixels, but of all stabilities combined. Figures (c) show percentage of repeated regions where each of the curves was scaled to 100% for the number of all detected regions of each ordering, while figures (d) show absolute region counts. For the boat sequence, where the images are monochromatic, the only pixel ordering yielding any regions is the ordering by the intensity. The results in bottom rows show that the achieved alignment is better for regions with higher stability. Here, all pixel orderings are combined. Again, left columns (e) show percentage of detected regions while right columns (f) show absolute region counts.

From the results on Graffiti and Bricks sequences we see that no pixel ordering is universally better than another, although the projection onto the intensity axis is generally a good choice. This observation is in agreement with the idea of run-time adaptation, described in Section 3.2.3. We also see that the range of thresholds where the region was unchanging represents a good estimation of the region stability – the measured alignment generally increased with increasing stability of the region. But, unfortunately, the absolute number of more stable regions rapidly decreases. In our experiments, we typically take all regions that are stable for 12 thresholds or more.

6.2.2 LAFs Repeatability

The evaluation of repeatability of construction of local affine frames follows the same protocol as the MSER evaluation in previous section. The same scenes were used, i.e. planar scenes for which the homographies between individual images are known. Alignment $a$ between frames from two images $I_1$ and $I_2$ is computed in the following way (see Figure 6.10 for reference). For each frame a square region $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$ is constructed (in the frame coordinates), which translates to a parallelogram in the image domain. Parallelograms detected in reference image $I_1$ are projected by the known homography $H_{I_1 \rightarrow I_2}$ into $I_2$. Let us denote $\Omega_1$ a parallelogram of a frame detected in $I_1$ and $\Omega_2$ a parallelogram of a frame detected in $I_2$. $\Omega'_1 = H_{I_1 \rightarrow I_2} \Omega_1$ is the parallelogram $\Omega_1$ projected from $I_1$ to $I_2$ by $H_{I_1 \rightarrow I_2}$. The alignment is then expressed as $a = \frac{|\Omega'_1 \cap \Omega_2|}{|\Omega_1'|}$, where $|\ldots|$ denotes the region area. For each LAF projected from the reference image ($I_1$), a single best-aligned LAF is selected on each of the remaining images in the set.

Figures 6.11, 6.12, and 6.13 show results of experimental evaluation of the frame alignment. As in the MSER evaluation, in (b) the alignment is shown for different viewpoints, with frames of all types combined. Figures (c) and (d) show the alignments of frames of individual types.
Figure 6.7: MSER evaluation results on Graffiti sequence.
6.2 Evaluation of Components of the Recognition System

Figure 6.8: MSER evaluation results on Boat sequence. As the images are monochromatic, the only pixel ordering yielding regions is by intensity.
Figure 6.9: MSER evaluation results on Bricks sequence.
6.2 Evaluation of Components of the Recognition System

Figure 6.10: Computation of LAF alignment. Two corresponding frames are detected in two images $I_1$ and $I_2$. Frame from $I_1$ is projected by a known homography $H_{I_1 \rightarrow I_2}$ to image $I_2$, and the alignment is given as the ratio of intersection and union of parallelograms corresponding to the frames.

Figures (c) show percentage of repeated frames where each of the curves is scaled to 100% for the number of all constructed frames of the particular type, while Figures (d) show absolute frame counts.

We see that generally the constructions based on covariance matrix are more stable than those based on tangent points. The covariance matrix, and the centre of gravity as well, are as integral quantities less influenced by local imprecisions in detection of region shape.

6.2.3 Reduction of the Number of LAFs

This experiment evaluates the impact of reduction of redundancy in image representation to recognition performance of the system. The reduction procedure is described in Section 3.5.1. Several runs on identical datasets were performed. The only difference in the whole recognition system was the configuration of the LAF reduction algorithm, which was set so that the number of query frames was reduced to 50%, 25%, resp. 10% of the number of detected frames. On database images all constructed frames were used.

The results show recognition performance of the complete system. All steps were executed, including the final consistency verification. A dominant plane is identified in the set of tentative correspondences (see Section 4.4), and inconsistent, out-of-the-plane correspondences are discarded. Every query image, portraying possibly more than one object, was matched against every database image. Thus $N \times M$ matching scores were obtained, where $N$, resp. $M$ is the number of query and database images respectively.

The recognition performance is presented in the form of a ROC curve which, parameterised by the matching score, shows the dependence between detection rate (true positives) and the number of false detections (false positives). The ROC curve is scaled by the number of query images, i.e. the false-positive rate is given as the average number of false detections per image. Correct localisation of detected objects was not checked. It is however very rare that the system identifies an object at wrong image location.

The results on the LAFsTest dataset (described in Section 6.1.5) are summarised in Table 6.1, and the ROC curve is shown in Figure 6.14 (a). In Figure 6.14 (b) a close-up of the operating part of the ROC curve is detailed, where the false detection rate is about 0.1 detections per image, i.e. where a false detection is reported on average on every tenth image. Table 6.1 states in its second column the percentage of query images where the correct object was identified in rank one, i.e. where the database images which obtained highest matching score represent (one of) the objects in the query image. Third column shows the average numbers of LAFs constructed on query images, it roughly corresponds to the configuration of the reduction algorithm. The representation build time in the fourth column gives the average time needed to build represen-
6 Experimental Validation

Figure 6.11: Evaluation of frame repeatability on the Graffiti sequence.

tation of a query image. The time includes MSER region detection, LAF construction, the LAF reduction, normalisation of measurement regions, and representation by a descriptor. The time needed for matching, i.e. for establishing of correspondences, is not included. See section 6.4 for discussion on computational complexity. Table 6.2 and Figure 6.15 present results for the ZuBuD dataset (Section 6.1.3) in the same form.

The experiment shows that unless the LAFs are reduced by a factor of 4 or more, the time saved by not computing the 75% of local representations is spent in the process deciding which frames to suppress. Note however, that the decision algorithm was not optimised for speed. Additional time is saved during the formation of tentative correspondences (Section 4.3), which is linear in the number of query frames. The matching time is however negligible, compared to representation building time, if the decision tree (Chapter 5) is used.

The recognition rates degrade gracefully with decreasing number of LAFs, up to again about 75% reduction. We conclude therefore, that the reduction increases the recognition speed significantly only if sequential matching is used – time is saved during the matching. If decision tree is used for establishing of local correspondences, the processing speed is increased if more that about 75% of frames are suppressed, but then the recognition performance deteriorates.
6.2 Evaluation of Components of the Recognition System

Figure 6.12: Evaluation of frame repeatability on the Boat sequence.

<table>
<thead>
<tr>
<th>LAFsTest: LAFs reduction</th>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100 %</td>
<td>95.29 %</td>
<td>2490</td>
<td>477 ms</td>
</tr>
<tr>
<td></td>
<td>50 %</td>
<td>92.94 %</td>
<td>1295</td>
<td>718 ms</td>
</tr>
<tr>
<td></td>
<td>25 %</td>
<td>90.00 %</td>
<td>647</td>
<td>419 ms</td>
</tr>
<tr>
<td></td>
<td>10 %</td>
<td>91.18 %</td>
<td>259</td>
<td>228 ms</td>
</tr>
</tbody>
</table>

Table 6.1: LAFsTest dataset: The impact of reduction of redundant representation.

<table>
<thead>
<tr>
<th>ZuBuD: LAFs reduction</th>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100 %</td>
<td>100.00 %</td>
<td>1341</td>
<td>265 ms</td>
</tr>
<tr>
<td></td>
<td>50 %</td>
<td>100.00 %</td>
<td>678</td>
<td>294 ms</td>
</tr>
<tr>
<td></td>
<td>25 %</td>
<td>97.39 %</td>
<td>338</td>
<td>223 ms</td>
</tr>
<tr>
<td></td>
<td>10 %</td>
<td>93.91 %</td>
<td>135</td>
<td>156 ms</td>
</tr>
</tbody>
</table>

Table 6.2: ZuBuD dataset: The impact of reduction of redundant representation.
6 Experimental Validation

Figure 6.13: Evaluation of frame repeatability on the Bricks sequence.

Figure 6.14: LAFsTest dataset: The impact of reduction of redundant representation on the recognition rate.
6.2 Evaluation of Components of the Recognition System

Figure 6.15: ZuBuD dataset: The impact of reduction of redundant representation on the recognition rate.

Figure 6.16: Impact of MSER boundary smoothing on recognition rate.

6.2.4 MSER Boundary Smoothing

This section examines the effect of smoothing of MSER regions (see Section 3.3.2, Figure 3.9) prior to detection of affine-covariant geometric primitives. Again, the system processed several times the same dataset. The configurations was identical except for the varying parameter $k$ of

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>No smoothing</td>
<td>94.12 %</td>
<td>3766</td>
<td>370 ms</td>
</tr>
<tr>
<td>$k = 10$</td>
<td>95.29 %</td>
<td>2514</td>
<td>317 ms</td>
</tr>
<tr>
<td>$k = 30$</td>
<td>94.71 %</td>
<td>2490</td>
<td>311 ms</td>
</tr>
<tr>
<td>$k = 50$</td>
<td>96.47 %</td>
<td>2302</td>
<td>282 ms</td>
</tr>
</tbody>
</table>

Table 6.3: LAFsTest dataset: MSER boundary smoothing.
6 Experimental Validation

ZuBuD: Region boundary smoothing

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>No smoothing</td>
<td>100.00 %</td>
<td>1743</td>
<td>185 ms</td>
</tr>
<tr>
<td>k = 10</td>
<td>100.00 %</td>
<td>1350</td>
<td>147 ms</td>
</tr>
<tr>
<td>k = 30</td>
<td>100.00 %</td>
<td>1341</td>
<td>143 ms</td>
</tr>
<tr>
<td>k = 50</td>
<td>100.00 %</td>
<td>1146</td>
<td>125 ms</td>
</tr>
</tbody>
</table>

Table 6.4: ZuBuD dataset: MSER boundary smoothing.

The measured ROC curves are shown in Figure 6.16. Since the complete curves look alike the curves in Figures 6.14 (a) and 6.15 (a) in previous section, only the close-ups are shown. Additional details of the experiments are summarised in Tables 6.3 and 6.4.

Stronger smoothing suppresses local contour structures, e.g. small concavities or curvature extrema, and therefore reduces the number of constructed frames. Consequently, the time needed to build the local representation is lower. From the ROC curves we see that the smoothed-out contour structures are not relevant for recognition, as the performance did not decrease. The performance is best for approximately \( k = 30 \), which is the value that we used in all the other experiments.

6.2.5 Required MSER Stability

Loosely speaking, stability of a MSER region is given as the number of consecutive thresholds over which the region does not significantly change (see Section 3.2.1). In Section 6.2.1 it was experimentally shown, that MSERs unchanging for more thresholds are more repeatable. Here we evaluate how the recognition rate is affected. Since the repeatability increases monotonically with stability, there is no benefit in using regions with lower stability while not using regions with higher. The regions are therefore selected by a “minimal” required stability, i.e. for a
6.2 Evaluation of Components of the Recognition System

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 and more</td>
<td>93.53 %</td>
<td>4258</td>
<td>478 ms</td>
</tr>
<tr>
<td>10 and more</td>
<td>93.53 %</td>
<td>3216</td>
<td>380 ms</td>
</tr>
<tr>
<td>12 and more</td>
<td>95.29 %</td>
<td>2490</td>
<td>318 ms</td>
</tr>
<tr>
<td>15 and more</td>
<td>94.71 %</td>
<td>1873</td>
<td>256 ms</td>
</tr>
<tr>
<td>20 and more</td>
<td>92.94 %</td>
<td>1201</td>
<td>182 ms</td>
</tr>
</tbody>
</table>

Table 6.5: LAFsTest dataset: Minimal required MSER stability.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 and more</td>
<td>100.00 %</td>
<td>2636</td>
<td>280 ms</td>
</tr>
<tr>
<td>10 and more</td>
<td>100.00 %</td>
<td>2118</td>
<td>243 ms</td>
</tr>
<tr>
<td>12 and more</td>
<td>100.00 %</td>
<td>1742</td>
<td>209 ms</td>
</tr>
<tr>
<td>15 and more</td>
<td>100.00 %</td>
<td>1341</td>
<td>170 ms</td>
</tr>
<tr>
<td>20 and more</td>
<td>100.00 %</td>
<td>847</td>
<td>124 ms</td>
</tr>
</tbody>
</table>

Table 6.6: ZuBuD dataset: Minimal required MSER stability.

given stability, all regions with higher stability are also used\(^2\).

The ROC curves are shown in Figure 6.17; Tables 6.5 and 6.6 show the number of LAFs constructed. The performance is poor if region stability 20 or higher is required, because the number of detected regions is too low. When regions of lower stability are added, the number of LAFs quickly increases, and the performance improves. However, the lower the stability of added regions, the lower is their added value. On the LAFsTest dataset, regions stable for less than 10 thresholds actually decreased the performance by introducing a large number of mismatches. We typically configure the system to use only regions stable for more than 12 or 15 thresholds.

6.2.6 Measurement Region Size

In this section we focus on the choice of size of Measurement Region (MR). Measurement region (see Section 4.1) is the part of image, defined in terms of an affine frame, whose appearance is encoded into descriptor and then used for matching. We use square MRs of different sizes, centred on a detected affine frame \( \mathbf{F} \) (see Figure 6.19 for illustration). Figure 6.18 shows the ROC curves obtained for four different MR sizes. For the LAFsTest dataset, the optimal size is \( (-1, 2) \times (-1, 2) \), which is our default configuration. On the ZuBuD dataset different MR sizes do not make much difference, since the objects of interest are virtually planar, and there is little occlusion. Therefore even larger MRs do not often extend to image areas outside objects.

6.2.7 Discretisation of Measurement Region

In this section we look into the problem of resolution of normalised patches. A measurement region, which is a parallelogram in input image, is rasterised into a canonical form (step 2 of Algorithm 3 in Section 4.1), i.e. to a square grid of pixels. What is the optimal resolution of the grid? Smaller resolutions are faster to compute, but for large image regions, the quality of represented image content is degraded. On the other hand, for small image regions, higher

\(^2\)In the binaries available at http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html, the minimal stability is referred to as 'minimum margin', command line option \(-\text{mm}\)
Figure 6.18: Recognition rate for variable size of Measurement Region. (a) LAFsTest dataset, (b) ZuBuD dataset.

Figure 6.19: Different sizes of measurement regions.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5x5</td>
<td>92.35 %</td>
<td>2490</td>
<td>186 ms</td>
</tr>
<tr>
<td>11x11</td>
<td>94.12 %</td>
<td>2490</td>
<td>255 ms</td>
</tr>
<tr>
<td>15x15</td>
<td>95.29 %</td>
<td>2490</td>
<td>324 ms</td>
</tr>
<tr>
<td>21x21</td>
<td>95.29 %</td>
<td>2490</td>
<td>473 ms</td>
</tr>
<tr>
<td>31x31</td>
<td>95.29 %</td>
<td>2490</td>
<td>845 ms</td>
</tr>
<tr>
<td>41x41</td>
<td>95.29 %</td>
<td>2490</td>
<td>1337 ms</td>
</tr>
<tr>
<td>51x51</td>
<td>95.29 %</td>
<td>2490</td>
<td>2088 ms</td>
</tr>
<tr>
<td>61x61</td>
<td>94.71 %</td>
<td>2490</td>
<td>2803 ms</td>
</tr>
<tr>
<td>101x101</td>
<td>95.29 %</td>
<td>2490</td>
<td>11112 ms</td>
</tr>
</tbody>
</table>

Table 6.7: LAFsTest dataset: Discretisation of measurement region, DCT representation.
6.2 Evaluation of Components of the Recognition System

Figure 6.20: Recognition rate for variable discretisation of Measurement Region using DCT representation. (a) LAFsTest dataset, (b) ZuBuD dataset.

Figure 6.21: Recognition rate for variable discretisation of Measurement Region using SIFT representation. (a) LAFsTest dataset, (b) ZuBuD dataset.
Experimental Validation

Figure 6.22: Recognition rate for variable discretisation of Measurement Region using RGB raster representation. LAFsTest dataset.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5x5</td>
<td>91.76 %</td>
<td>2490</td>
<td>217 ms</td>
</tr>
<tr>
<td>11x11</td>
<td>93.53 %</td>
<td>2490</td>
<td>322 ms</td>
</tr>
<tr>
<td>15x15</td>
<td>94.12 %</td>
<td>2490</td>
<td>439 ms</td>
</tr>
<tr>
<td>21x21</td>
<td>94.71 %</td>
<td>2490</td>
<td>699 ms</td>
</tr>
<tr>
<td>31x31</td>
<td>95.88 %</td>
<td>2490</td>
<td>1278 ms</td>
</tr>
<tr>
<td>41x41</td>
<td>95.29 %</td>
<td>2490</td>
<td>2107 ms</td>
</tr>
<tr>
<td>51x51</td>
<td>95.29 %</td>
<td>2490</td>
<td>3142 ms</td>
</tr>
<tr>
<td>61x61</td>
<td>95.29 %</td>
<td>2490</td>
<td>11581 ms</td>
</tr>
<tr>
<td>101x101</td>
<td>94.71 %</td>
<td>2490</td>
<td>1581 ms</td>
</tr>
</tbody>
</table>

Table 6.8: LAFsTest dataset: Discretisation of measurement region, SIFT representation.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5x5</td>
<td>88.82 %</td>
<td>2490</td>
<td>180 ms</td>
</tr>
<tr>
<td>11x11</td>
<td>91.18 %</td>
<td>2490</td>
<td>241 ms</td>
</tr>
<tr>
<td>15x15</td>
<td>91.18 %</td>
<td>2490</td>
<td>303 ms</td>
</tr>
<tr>
<td>21x21</td>
<td>90.59 %</td>
<td>2490</td>
<td>432 ms</td>
</tr>
<tr>
<td>31x31</td>
<td>88.82 %</td>
<td>2490</td>
<td>770 ms</td>
</tr>
<tr>
<td>41x41</td>
<td>91.18 %</td>
<td>2490</td>
<td>1173 ms</td>
</tr>
<tr>
<td>51x51</td>
<td>91.18 %</td>
<td>2490</td>
<td>1826 ms</td>
</tr>
<tr>
<td>61x61</td>
<td>89.41 %</td>
<td>2490</td>
<td>2426 ms</td>
</tr>
</tbody>
</table>

Table 6.9: LAFsTest dataset: Discretisation of measurement region, Raster representation.
6.2 Evaluation of Components of the Recognition System

### ZuBuD: Patch resolution, DCT representation

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5x5</td>
<td>100.00 %</td>
<td>1341</td>
<td>112 ms</td>
</tr>
<tr>
<td>11x11</td>
<td>100.00 %</td>
<td>1341</td>
<td>151 ms</td>
</tr>
<tr>
<td>15x15</td>
<td>100.00 %</td>
<td>1341</td>
<td>183 ms</td>
</tr>
<tr>
<td>21x21</td>
<td>100.00 %</td>
<td>1341</td>
<td>266 ms</td>
</tr>
<tr>
<td>31x31</td>
<td>100.00 %</td>
<td>1341</td>
<td>457 ms</td>
</tr>
<tr>
<td>41x41</td>
<td>100.00 %</td>
<td>1341</td>
<td>719 ms</td>
</tr>
<tr>
<td>51x51</td>
<td>100.00 %</td>
<td>1341</td>
<td>1114 ms</td>
</tr>
<tr>
<td>61x61</td>
<td>100.00 %</td>
<td>1341</td>
<td>1481 ms</td>
</tr>
<tr>
<td>101x101</td>
<td>100.00 %</td>
<td>1341</td>
<td>5482 ms</td>
</tr>
</tbody>
</table>

Table 6.10: ZuBuD dataset: Discretisation of measurement region, DCT representation.

### ZuBuD: Patch resolution, SIFT representation

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5x5</td>
<td>94.78 %</td>
<td>1341</td>
<td>315 ms</td>
</tr>
<tr>
<td>11x11</td>
<td>98.26 %</td>
<td>1341</td>
<td>330 ms</td>
</tr>
<tr>
<td>15x15</td>
<td>97.39 %</td>
<td>1341</td>
<td>364 ms</td>
</tr>
<tr>
<td>21x21</td>
<td>100.00 %</td>
<td>1341</td>
<td>552 ms</td>
</tr>
<tr>
<td>31x31</td>
<td>100.00 %</td>
<td>1341</td>
<td>843 ms</td>
</tr>
<tr>
<td>41x41</td>
<td>100.00 %</td>
<td>1341</td>
<td>1328 ms</td>
</tr>
<tr>
<td>51x51</td>
<td>100.00 %</td>
<td>1341</td>
<td>1791 ms</td>
</tr>
<tr>
<td>61x61</td>
<td>100.00 %</td>
<td>1341</td>
<td>2463 ms</td>
</tr>
<tr>
<td>101x101</td>
<td>100.00 %</td>
<td>1341</td>
<td>6245 ms</td>
</tr>
</tbody>
</table>

Table 6.11: ZuBuD dataset: Discretisation of measurement region, SIFT representation.

resolution results only in storing a higher number of interpolated pixels. The optimum proved to be depending on the type of descriptor which is subsequently used to represent the patch. The experiment was therefore repeated three times for different descriptors: DCT, SIFT, and “raster”. The “raster” representation consists directly from the normalised pixels and cross-correlation is used as the similarity measure.

For the DCT descriptor, the ROC curves are shown in Figure 6.20. Starting from patch resolution of 11 × 11 pixels, the performance is more or less stable. Performance gain of higher resolution is minimal, or none at all. This observation corresponds with the nature of the DCT representation, which stores only low-frequency coefficients. Higher resolution of normalised patches increases energy in higher frequencies, while the low frequencies are almost unaffected. See also corresponding Tables 6.7 and 6.10. The most important figure here is the time required for computation of the image representation. Several computation steps depends linearly on the number of pixels in normalised patches: bilinear interpolation of image pixels, computation of photometric transformation and photometric normalisation, and computing the descriptor as dot-products with DCT basis (Equation 4.3).

Figure 6.21 shows the ROC curves for the SIFT descriptor. Here we see a different behaviour. The performance of SIFT increases with increasing patch resolution, especially on the ZuBuD dataset. Again, the observation is consistent with the SIFT nature, as SIFT consists of histograms of gradient orientations. At patch resolutions 51 × 51 and higher, the SIFT descriptor performs better (especially on ZuBuD) than the DCT descriptor. But as we see from Tables 6.8 and 6.11 the SIFT computation using 51 × 51 patches is more than 10 times slower than DCT computation on 11 × 11 patches.
6.2.8 Descriptors

Different descriptors of local appearance are compared here. The results, shown in Figures 6.23 and 6.24, are de facto just rearrangements of results from previous section for patch resolutions $11 \times 11$ and $21 \times 21$ pixels. Such a low resolution is chosen for computational reasons, but, as
6.2 Evaluation of Components of the Recognition System

Figure 6.24: Recognition rate for different descriptors of normalised appearance. MR discretisation $11 \times 11$ pixels. (a) LAFsTest dataset, (b) ZuBuD dataset.

<table>
<thead>
<tr>
<th>LAFsTest: Descriptors of normalised 11x11 patches</th>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>93.53 %</td>
<td>2490</td>
<td>322 ms</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>94.12 %</td>
<td>2490</td>
<td>255 ms</td>
<td></td>
</tr>
<tr>
<td>GreyDCT</td>
<td>94.12 %</td>
<td>2490</td>
<td>253 ms</td>
<td></td>
</tr>
<tr>
<td>Raster</td>
<td>91.18 %</td>
<td>2490</td>
<td>241 ms</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.13: LAFsTest dataset: Choice of descriptor of normalised appearance. Discretisation 11x11 pixels.

<table>
<thead>
<tr>
<th>ZuBuD: Descriptors of normalised 21x21 patches</th>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>100.00 %</td>
<td>1341</td>
<td>552 ms</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>100.00 %</td>
<td>1341</td>
<td>266 ms</td>
<td></td>
</tr>
<tr>
<td>GreyDCT</td>
<td>100.00 %</td>
<td>1341</td>
<td>263 ms</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.14: ZuBuD dataset: Choice of descriptor of normalised appearance. Discretisation 21x21 pixels.

<table>
<thead>
<tr>
<th>ZuBuD: Descriptors of normalised 11x11 patches</th>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>98.26 %</td>
<td>1341</td>
<td>330 ms</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>100.00 %</td>
<td>1341</td>
<td>151 ms</td>
<td></td>
</tr>
<tr>
<td>GreyDCT</td>
<td>100.00 %</td>
<td>1341</td>
<td>146 ms</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.15: ZuBuD dataset: Choice of descriptor of normalised appearance. Discretisation 11x11 pixels.
Experimental Validation

Figure 6.25: Ordering of DCT coefficients (Equation 4.3) by increasing frequency.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Diagonals</td>
<td>89.41 %</td>
<td>2490</td>
<td>360 ms</td>
</tr>
<tr>
<td>3 Diagonals</td>
<td>93.53 %</td>
<td>2490</td>
<td>385 ms</td>
</tr>
<tr>
<td>4 Diagonals</td>
<td>95.29 %</td>
<td>2490</td>
<td>404 ms</td>
</tr>
<tr>
<td>5 Diagonals</td>
<td>94.71 %</td>
<td>2490</td>
<td>445 ms</td>
</tr>
<tr>
<td>6 Diagonals</td>
<td>95.29 %</td>
<td>2490</td>
<td>482 ms</td>
</tr>
<tr>
<td>8 Diagonals</td>
<td>95.29 %</td>
<td>2490</td>
<td>589 ms</td>
</tr>
<tr>
<td>10 Diagonals</td>
<td>94.71 %</td>
<td>2490</td>
<td>713 ms</td>
</tr>
<tr>
<td>12 Diagonals</td>
<td>94.71 %</td>
<td>2490</td>
<td>865 ms</td>
</tr>
<tr>
<td>15 Diagonals</td>
<td>94.71 %</td>
<td>2490</td>
<td>1134 ms</td>
</tr>
<tr>
<td>20 Diagonals</td>
<td>93.53 %</td>
<td>2490</td>
<td>1739 ms</td>
</tr>
</tbody>
</table>

Table 6.16: LAFsTest dataset: Number of DCT coefficients used for representation of local appearance. Discretisation 21x21 pixels.

shown in previous section, is not particularly fair to the SIFT descriptor, which requires higher resolutions for optimal performance. The results here also include another descriptor, “Grey-DCT”, which is the DCT representation of only intensity values of the normalised patches. This representation does not equal to using greyscale images though, the photometric normalisation and computation and verification of photometric region-to-region transformations is still in RGB space.

The results are shown in Figures 6.23 and 6.24, and in Tables 6.12, 6.13, 6.14 and 6.15. Again, the raster representation is not shown on the ZuBuD database, because of its memory requirements. We see that the DCT representation performs best and is fast to compute.

To conclude the results from this and from the previous sections, a reasonable computational and performance compromise is obtained by using DCT representation on 21 x 21 patches. This is also the most common configuration of the system. Faster computation can be achieved by reducing the resolution of the normalised patches, while slightly better performance by using SIFT descriptor on high resolution patches.
6.2 Evaluation of Components of the Recognition System

Figure 6.26: DCT representation: Recognition rate for different number of stored coefficients. MR discretisation $21 \times 21$ pixels. (a) LAFsTest dataset, (b) ZuBuD dataset.

Figure 6.27: DCT representation: Recognition rate for different number of stored coefficients. MR discretisation $11 \times 11$ pixels. (a) LAFsTest dataset, (b) ZuBuD dataset.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Diagonals</td>
<td>90.00 %</td>
<td>2490</td>
<td>220 ms</td>
</tr>
<tr>
<td>3 Diagonals</td>
<td>93.53 %</td>
<td>2490</td>
<td>224 ms</td>
</tr>
<tr>
<td>4 Diagonals</td>
<td>95.29 %</td>
<td>2490</td>
<td>232 ms</td>
</tr>
<tr>
<td>5 Diagonals</td>
<td>94.71 %</td>
<td>2490</td>
<td>244 ms</td>
</tr>
<tr>
<td>6 Diagonals</td>
<td>94.12 %</td>
<td>2490</td>
<td>274 ms</td>
</tr>
<tr>
<td>8 Diagonals</td>
<td>94.71 %</td>
<td>2490</td>
<td>292 ms</td>
</tr>
<tr>
<td>10 Diagonals</td>
<td>94.12 %</td>
<td>2490</td>
<td>334 ms</td>
</tr>
</tbody>
</table>

Table 6.17: LAFsTest dataset: Number of DCT coefficients used for representation of local appearance. Discretisation $11\times11$ pixels.
6 Experimental Validation

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Diagonals</td>
<td>99.13 %</td>
<td>1341</td>
<td>198 ms</td>
</tr>
<tr>
<td>3 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>209 ms</td>
</tr>
<tr>
<td>4 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>224 ms</td>
</tr>
<tr>
<td>5 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>245 ms</td>
</tr>
<tr>
<td>6 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>267 ms</td>
</tr>
<tr>
<td>8 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>322 ms</td>
</tr>
<tr>
<td>10 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>391 ms</td>
</tr>
<tr>
<td>12 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>470 ms</td>
</tr>
<tr>
<td>15 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>617 ms</td>
</tr>
<tr>
<td>20 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>935 ms</td>
</tr>
</tbody>
</table>

Table 6.18: ZuBuD dataset: Number of DCT coefficients used for representation of local appearance. Discretisation 21x21 pixels.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Correct in rank 1</th>
<th>Avg N of frames</th>
<th>Avg representation build time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Diagonals</td>
<td>99.13 %</td>
<td>1341</td>
<td>129 ms</td>
</tr>
<tr>
<td>3 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>132 ms</td>
</tr>
<tr>
<td>4 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>137 ms</td>
</tr>
<tr>
<td>5 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>143 ms</td>
</tr>
<tr>
<td>6 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>153 ms</td>
</tr>
<tr>
<td>8 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>171 ms</td>
</tr>
<tr>
<td>10 Diagonals</td>
<td>100.00 %</td>
<td>1341</td>
<td>190 ms</td>
</tr>
</tbody>
</table>

Table 6.19: ZuBuD dataset: Number of DCT coefficients used for representation of local appearance. Discretisation 11x11 pixels.

6.2.9 Number of Low-Frequency DCT Coefficients

As in the JPEG compression scheme, the coefficients of the matrix $D$ (see Equation 4.3) are ordered diagonally by increasing frequency. An $i$-th (minor) diagonal is composed of such coefficients, where indices $p + q = i - 1$. See Figure 6.25 for illustration. Here we evaluate what is the optimal number of the diagonals, from which the DCT descriptor is composed. The first diagonal, the single top-left coefficient, is not used. It represents the mean value of the patch, which, due to photometric normalisation, is constant for all frames, and therefore provide no discrimination.

The experiment was repeated for patch resolutions $11 \times 11$ and $21 \times 21$ pixels. The number of coefficients was increased one diagonal at a time, up to the patch resolution. The results, shown in Figures 6.26 and 6.27, and in Tables 6.16, 6.17, 6.18 and 6.19, are interesting. On LAFsTest dataset, 4 or 5 diagonals (14 coefficients) gives the best rates. On ZuBuD, the optimum is at as little as 2 or 3 diagonals (2 or 5 coefficients). This is due to planarity of the ZuBuD objects. The dominant plane found during the search for geometrically consistent tentative correspondences (Section 4.4) has significantly larger support than planes formed from random configuration of mismatches. On ZuBuD dataset, a reasonably reliable recognition can be achieved by geometric configuration of detected LAFs alone.
6.3 Performance Comparison with Published Results

This section compares recognition performance of our system to other results published in the literature. The evaluation is made on publicly available databases, COIL-100 (see Section 6.1.2), ZuBuD (see Section 6.1.3), and Focus (see Section 6.1.4).

### 6.3.1 Recognition Performance on COIL-100 Dataset

The COIL-100 dataset has been widely used in assessing performance of object recognition methods. Table 6.20 compares the recognition rates achieved by our system with results of other published methods. Results are presented for five experimental set-ups, differing in the number of training views per object. Decreasing the number of training views increases demands on the method’s generalisation ability, and on the insensitivity to object appearance deformations. The LAF approach performs best in all experiments, regardless of the number of training views. For only four training views per object (90° apart, 68 test views per object), the recognition rate is over 98%, demonstrating the remarkable robustness to changes in viewpoint. In the case of 18 training views per object, only 5 out of the total 5400 test images were misclassified.

Table 6.21 provides detailed information about the experiments. Two variants of the recognition system were evaluated, one which recalls the stored frames via the proposed decision tree (with sub-linear recall time, see Chapter 5), and a second one which sequentially scans through all stored frames (linear recall time). The recall times in the Table 6.21 show that using the decision tree, matching of approximately 500 query frames against hundreds of thousands of stored frames takes about 2 milliseconds. The total response time of the recognition system is the sum of the time needed to build the query image representation (independent of the number of database objects – 6th row of Table 6.21) and the recall time (7th or 8th row). Note that doubling the number of training images (columns 2 and 3) did not double the recall time for the tree approach. The required time increased from 1.99 ms to 2.17 ms, i.e. by less than 10%. It confirms the claim that the recall time is sub-linear in the number of stored frames. Training of the tree took approximately 30 hours and the tree representation required approximately 1GB of memory.

### 6.3.2 Recognition Performance on FOCUS Dataset

On the FOCUS dataset we run an experiment with identical setup as in evaluation of the SEDL system introduced by Cohen [Coh99]. The quality of the retrieval is assessed by the same two
### 6 Experimental Validation

<table>
<thead>
<tr>
<th>MSER+LAF</th>
<th>COIL-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Training view distance</td>
<td>90° 45°</td>
</tr>
<tr>
<td>2. Number of DB images</td>
<td>400 800</td>
</tr>
<tr>
<td>3. Number of DB frames</td>
<td>186346 385197</td>
</tr>
<tr>
<td>4. Number of query images</td>
<td>6800 6400</td>
</tr>
<tr>
<td>5. Avg number of query frames</td>
<td>494 494</td>
</tr>
<tr>
<td>6. Avg time to build representation</td>
<td>218 ms 221 ms</td>
</tr>
<tr>
<td>7. Avg recall time without the tree</td>
<td>293 ms 697 ms</td>
</tr>
<tr>
<td>8. Avg recall time with the tree</td>
<td>1.99 ms 2.17 ms</td>
</tr>
<tr>
<td>9. Recognition rate</td>
<td>98.24% 99.77%</td>
</tr>
</tbody>
</table>

Table 6.21: COIL-100: Experimental results.

![Examples of query (left) and corresponding database images (right) not retrieved.](image)

Figure 6.28: FOCUS: Examples of query (left) and corresponding database images (right) not retrieved.

quantities as defined by Cohen, the recall rate $r_R$ and the precision $\rho_R$:

$$
    r_R = \frac{n}{N} \\
    \rho_R = \frac{\sum_{i=1}^{n} (R + 1 - r_i)}{\sum_{i=1}^{n} (R + 1 - i)}
$$

(6.1)

where $n$ is the number of correct answers in the first $R$ retrieved images, $N$ the number of all correct answers contained in the database, and $r_i$ is the rank of the $i$-th correctly retrieved answer.

In Table 6.22, average recall rate $r_{20}$ and average precision $\rho_{20}$ are given for the number of retrieved images $R = 20$. For each of the 25 queries used by Cohen, the database images were sorted according to the matching score, and the recall $r_{20}$ and the precision $\rho_{20}$ were computed according to Equation 6.1. Each of the 25 queries has 2 to 9 correct answers in the database, with the total number of all correct answers equal to 90. Our method achieves a 83% recall, which is approximately 5% better than results reported by Cohen. Most database images missed depict objects different from the query. Figure 6.28 shows three such examples. The “failure” in such cases might be viewed as a strength, demonstrating the very high selectivity of the method, distinguishing items that superficially look identical.

<table>
<thead>
<tr>
<th>SEDL</th>
<th>LAFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>recall $r_{20}$</td>
<td>avg precision $\rho_{20}$</td>
</tr>
<tr>
<td>70/90 = 77.8%</td>
<td>88%</td>
</tr>
</tbody>
</table>

Table 6.22: FOCUS: Retrieval performance compared to the SEDL system.

#### 6.3.3 Recognition Performance on ZuBuD Dataset

With the ZuBuD dataset (see Section 6.1.3), 115 query images were matched against 1005 database images, ie. 115575 matches were evaluated in total. For every query image, the $R$
### 6.4 Analysis of Computational Complexity

<table>
<thead>
<tr>
<th>MSER+LAF</th>
<th>ZuBuD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Number of DB images</td>
<td>1005</td>
</tr>
<tr>
<td>2. Number of DB frames</td>
<td>251633</td>
</tr>
<tr>
<td>3. Number of query images</td>
<td>115</td>
</tr>
<tr>
<td>4. Avg number of query frames</td>
<td>1341</td>
</tr>
<tr>
<td>5. Avg time to build representation</td>
<td>267 ms</td>
</tr>
<tr>
<td>6. Avg recall time without tree</td>
<td>27234 ms</td>
</tr>
<tr>
<td>7. Avg recall time with tree</td>
<td>14.3 ms</td>
</tr>
<tr>
<td>8. Recognition rate without tree</td>
<td>100%</td>
</tr>
<tr>
<td>9. Recognition rate with tree</td>
<td>93%</td>
</tr>
</tbody>
</table>

**Other Methods**

| HPAT [SSTG03b] recognition rate | 86.1% |
| Random subwindows [MGPW05] recognition rate | 95.7% |

Table 6.23: ZuBuD: Experimental results.

Closest database images were retrieved. The recall rate \( r_R \) was evaluated, which is defined as \( r_R = \frac{n_R}{N} \), where \( n_R \) is the number of correct answers in the first \( R \) retrieved images, and \( N \) the number of all possible correct answers. For ZuBuD, where every query has 5 corresponding images in the database, \( N = \min(R, 5) \).

Details on the experiment are given in Table 6.23. Again, two variants of the system were evaluated, one with sub-linear recall time using the decision tree (see Chapter 5), and a second one which sequentially scans through all stored frames with linear recall time. The slower tree recall times, compared to the COIL-100 dataset, are caused by a higher number of query frames per image, and by the increase of the tree leaf capacity from 4 to 10 – up to 10 frames were searched exhaustively in the leaf nodes (refer to Section 5.2.2). The leaf capacity represents a trade-off between recall speed and recognition rate. Setting the capacity to 1000, a recognition rate of 98.2% was achieved, but the average recall time dropped to 510 ms. Linear exhaustive scan through all the stored frames (avoiding the tree) achieved recognition rate of 100%, but with recall times over 27 seconds per image. Bottom of Table 6.23 shows results published by others.

#### 6.4 Analysis of Computational Complexity

Figure 6.29 presents analysis of computational demands of components of the recognition system. When a query image is processed, representation of the image is built first. It involves MSER region detection (Section 3.2.1), detection of affine-covariant geometric primitives (Section 3.3), construction of Local Affine Frames (Section 3.4), computation of geometrically and photometrically normalised representation of local appearance (Section 4.1), and DCT transformation (Section 4.2). In the second step, matching of the query representation against database, the computation time varies depending on the database size. When the sequential matching was used, the major computational bottleneck of the system was, even for rather small databases (tens of objects), in the matching stage. Introducing the decision tree (Chapter 5), the matching time was reduced to a few milliseconds (for hundreds of objects), and the bottleneck moved to the process of local patch normalisation and computation of DCT representation. There is not much to be improved algorithmically, the computational load is given by the amount of data being processed, i.e. the number of patches and their resolution. Reduction of the number of patches was attempted in Section 3.5 with unconvincing results (Section 6.2.3), and the problem of selection of patch resolution was examined in Section 6.2.7.
Significant performance improvement was achieved by implementing the image processing part in graphics card hardware. The implementation includes geometric and photometric normalisation of the patches, and computation of the DCT representation. Processing patches from a single query image (few thousands of patches) takes 10–20 milliseconds, which is roughly 10 times faster than in standard CPU implementation.

On a modern PC, as of fall 2006, our recognition system is able of recognising objects from a database of circa 50 objects at speed of about 8 frames per second.
7 Object Recognition and Colour Constancy

In this chapter we show an application of the MSER-LAF recognition method to the problem of colour constancy. We demonstrate that even under severe changes of illumination, many objects are reliably recognised if relying only on geometry and on invariant representation of local colour appearance. We feel that colour constancy as a preprocessing step of an object recognition algorithm is important only in cases when colour is major, or the only available, clue for object discrimination. We also show that successful object recognition allows for “colour constancy by recognition” – an approach where the global photometric transformation is estimated from locally corresponding image elements.

7.1 Interaction Between Object Recognition and Colour Constancy

Colour constancy is a classical problem that has been recently connected to object recognition [SB91, FBM98, BFM00, SH96]. In [FBM98], Funt et al. propose to judge the quality of colour constancy algorithms by their impact on recognition rates. The question “Is colour constancy good enough (for object recognition)?” is posed. For histogram intersection as the recognition method and a wide range of colour constancy algorithms their answer is negative, i.e. none of the tested colour constancy algorithms is “good enough”.

We revisit the issue and show that if a recognition method relies mainly on geometry and representation of local colour appearance invariant to affine transformation of colour components (equivalent to a diagonal colour constancy model [FDF94] with an offset term), object recognition can be successful even under severe and unknown change of illumination. This is experimentally demonstrated on a public dataset from the Simon Fraser University, that has been previously used in colour constancy experiments [Bar, BFM00].

Successful recognition insensitive to illumination allows us to consider the intuitive approach of “colour constancy by recognition”. We show experimentally that a straightforward approach which estimates the colour transformation from local correspondences established in the recognition step is more precise than the best standard (global, correspondence-less) colour constancy method. The precision is measured by the distance (in the chromatic plane) of the white point under canonical illumination and transformed white point of image under the unknown illumination. The achieved precision is approximately three times higher than that of Barnard et al. [BFM00].

The result has to be interpreted carefully. Clearly, the presence of a known object in the scene is a restrictive assumption. Colour constancy is often required in scenes without known object, e.g. as a part of a white balance module of a camera. The message is rather that if a known object, or object class (hair, skin), is in the scene, much better results of colour constancy can be expected, if the object is recognised. It seems that two different classes of colour constancy algorithms might be distinguished: those relying on global or statistical properties and those attempting to recognise objects or object classes and use constraints on scene illumination imposed by observed colours of known surfaces. Unlike the former, the later colour constancy algorithms are able to deal with non-uniform illumination. In a synthetic experiments, we show that it is possible to partition the image according to the illuminant.
7.2 Finding Global Photometric Transformation

The object appearance representation described in Chapter 4 is invariant to affine extension of the monochromatic diagonal model. As depicted in Figure 4.1, the normalised appearance of corresponding patches is well matched even for images taken in very different illumination conditions. At local scale, such a simple photometric model is sufficient to establish correspondences. Global colour transformation is computed after the correspondences are found, using full affine model. By considering only the image regions that were put into correspondence, the global transformation is not influenced by background clutter or occluding objects.

From Patch-to-Patch to Pixel-to-Pixel Correspondences. Every established correspondence locally maps a pair of patches. Assuming that local geometric deformations are sufficiently well approximated by the 2D affine transformations, pixel correspondences are obtained by sampling the images with respect to the local coordinate systems of corresponding LAFs. This can be interpreted as a regular sampling of the geometrically normalised MRs depicted in Figure 4.1 (b) and (e). In our implementation, we sample the MRs on a regular $6 \times 6$ grid, obtaining thus 36 pixel-correspondences per every frame-correspondence. For a typical object, the number of such pixel-correspondences is in the order of thousands.

Computing Image-to-Image Photometric Transformation. With thousands of corresponding pixels available, the global query-to-database photometric transformation $P_{QD}$ can be calculated in a form more complex than diagonal without the risk of overfitting. We model the transformation as affine:

$$P_{QD}: \begin{pmatrix} r^D \\ g^D \\ b^D \end{pmatrix} = \begin{pmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & m_9 \end{pmatrix} \begin{pmatrix} r^Q \\ g^Q \\ b^Q \end{pmatrix} + \begin{pmatrix} n_r \\ n_g \\ n_b \end{pmatrix}.$$  \hspace{1cm} (7.1)

The transformation coefficients are obtained by least squares fitting, i.e. the sum of square differences between transformed colours of query pixels and colours of corresponding database pixels is minimised.

7.3 Experiments

Figure 7.1: Barnard’s dataset [Bar]: 20 database images.

Dataset. The experiments were conducted on a dataset that was made available by Kobus Barnard [Bar]. The dataset contains images of 20 different objects (Figure 7.1), every object
Figure 7.2: Examples of recognition failures. The failures are caused by the objects being presented from an unseen view, not due to different illumination.

Figure 7.3: A scene with multiple illuminants: (a) a synthetic query image, two differently illuminated halves joined, (b) found correspondences clustered by local photometric transformation, (c) corresponding database image.

is taken under 11 illuminants. The total number of images in the dataset is thus 220. The illuminants were chosen to cover the range of common illumination conditions. For each image, chromaticity of the white point is provided, which was obtained by temporarily placing a sheet of white paper in the scene.

The object recognition task is simplified by the fact that the objects are placed on a black background. The results of the recognition experiment are summarized in Table 7.1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSER-LAF</td>
<td>89.1 %</td>
</tr>
<tr>
<td>Histogram Intersection, no Colour Constancy</td>
<td>42.3%</td>
</tr>
<tr>
<td>Histogram Intersection, manual CC</td>
<td>87.7%</td>
</tr>
<tr>
<td>Histogram Intersection, best CC</td>
<td>80.9%</td>
</tr>
<tr>
<td>Histogram Intersection, worst CC</td>
<td>15.5%</td>
</tr>
</tbody>
</table>

Table 7.1: Summary of the recognition experiment. Recognition rate compared to other methods.
Table 7.2: Individual illuminants: Recognition rate and error of illuminant colour estimation.

<table>
<thead>
<tr>
<th>Illuminant</th>
<th>Recognition rate</th>
<th>White Point error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ph-ulm</td>
<td>17/20, 85%</td>
<td>0.015</td>
</tr>
<tr>
<td>solux-3500+3202</td>
<td>19/20, 95%</td>
<td>0.011</td>
</tr>
<tr>
<td>solux-3500</td>
<td>19/20, 95%</td>
<td>0.006</td>
</tr>
<tr>
<td>solux-4100+3202</td>
<td>17/20, 85%</td>
<td>0.013</td>
</tr>
<tr>
<td>solux-4100</td>
<td>20/20, 100%</td>
<td>0.008</td>
</tr>
<tr>
<td>solux-4700+3202</td>
<td>12/20, 60%</td>
<td>0.021</td>
</tr>
<tr>
<td>solux-4700</td>
<td>19/20, 95%</td>
<td>0.012</td>
</tr>
<tr>
<td>syl-50MR16Q+3202</td>
<td>18/20, 90%</td>
<td>0.009</td>
</tr>
<tr>
<td>syl-50MR16Q</td>
<td>20/20, 100%</td>
<td>–</td>
</tr>
<tr>
<td>syl-cwf</td>
<td>16/20, 80%</td>
<td>0.010</td>
</tr>
<tr>
<td>syl-wwf</td>
<td>19/20, 95%</td>
<td>0.013</td>
</tr>
<tr>
<td>average</td>
<td>89%</td>
<td>0.012</td>
</tr>
<tr>
<td>best method in [BFM00]</td>
<td>81%</td>
<td>0.038</td>
</tr>
</tbody>
</table>

Experimental Protocol. The training database (the set of known images) contains a single image of every object – images taken under ‘syl-50MR16Q’ illuminant were used. All the database images are shown in Figure 7.1. To follow the experimental setup from [BFM00], all 220 images are used as queries, i.e. the set of queries contains also the database images. Every query image is matched against every database image. As there are no images of non-database objects, the database image with the highest score is always selected (forced match).

For the estimation of colour constancy we have manually selected only those query–database image pairs where the object was successfully recognised. Global photometric transformation with 12 degrees of freedom was estimated from corresponding regions, as described in Section 7.2. Note that the query-to-database photometric transformation can not be used to estimate directly the colour of the illuminant, since image taken under “white light” are not part of the database.

The precision of the estimated global photometric transformation is verified by transforming the provided white paper colour in the query images with $P_{QD}$. Ideally, the transformed colour should equal to the white paper colour of the matched database image. As it does not, the precision of the estimate is measured by computing the euclidean distance between chromaticities of the transformed query white point and the database white point.

Results. Results of the recognition experiments are summarised in Table 7.1. Our method (MSER-LAF) is compared to results published in [BFM00]. In [BFM00], query images are first adjusted by one of a rather exhaustive set of 23 colour constancy algorithms. Matching is then done on the adjusted images by the histogram intersection method.

The first row of Table 7.1 shows the recognition rate of our method, second row of the histogram intersection method without any colour constancy being applied. The third row shows results for manual colour constancy, where the query images were transformed so that the manually measured white points match. The remaining two rows report results for the best “non-diagonal, coefficient-rule” and the worst “color-in-perspective” of the 23 colour constancy algorithms. Our recognition performance is superior to any of the results presented in [BFM00].

Table 7.2 shows how individual illuminants affect recognition rate of the MSER-LAF method. There is no significant difference in the performance, except for the ‘solux-4700+3202’ illuminant.
(4700K incandescent light plus a blue filter). The recognition does not fail here due to the illuminant colour, but due to the low intensity of images captured under this lighting. The third column of Table 7.2 shows the precision of the global photometric transformation estimation. For comparison, a white point estimation error of the best performing method from [BFM00] is quoted. Our estimates are on average three times more precise, but note that only correctly recognised images are included. Estimation based on mismatched objects may produce arbitrary photometric transformation.

Figure 7.2 shows all our recognition failures in queries for the first four database objects. The query images differ from the database images not only in the illumination, but, more significantly, in the object pose. The balls are rotated so that their visual appearance is substantially different from the database images. The blocks-object was turned upside-down, producing a ‘mirror’ image of itself, which is not recognised by our method. Refer to Figure 7.1 to see the differences between database images and the unrecognised queries.

Multiple Illuminants. In a final experiment, we demonstrate that our recognition system can handle objects viewed under multiple illuminants at the same time, as can be the case e.g. when a shadow is cast over part of an object. Figure 7.3 (a) shows a synthetic query image, which was obtained by artificially merging two differently illuminated images of the object. The process of image description and matching (Chapter 4) is invariant to local illumination. The presence of multiple illuminants have thus no effect on the obtained correspondences, except for LAFs that are located on the boundary of differently illuminated object areas.

Obtained correspondences are clustered by their local photometric transformation. Each such cluster represents a global transformation caused by one of the illuminants. In Figure 7.3 (b) two clusters of correspondences are shown with green and white dots respectively. With a single exception, the correspondences are correctly separated according to the illuminant.

7.4 Concluding Remarks

We have revisited the connection between colour constancy and object recognition. For many objects a recognition method relying mainly on geometry and on photometrically invariant representation of local appearance can be successful even under severe and unknown changes of illumination. Successful object recognition allows for “colour constancy by recognition” – an approach where the global photometric transformation is estimated from locally corresponding image patches. Since the recognition method is insensitive to object occlusion and background clutter, the colour constancy by recognition approach is successful even in situations when the known object occupy only a small portion of otherwise unknown scene.

Experimentally we have shown that our recognition method outperforms the methods described in [BFM00], i.e. the histogram intersection algorithm applied after colour correction. The recognition rate was virtually independent of the illuminant, changes in objects’ poses had a much stronger impact on the results. When an object was correctly recognised, even a straightforward least-squares algorithm was able to estimate the global photometric transformation three times more precisely than the best correspondence-less colour constancy method published in [BFM00]. An experiment on a scene where different parts of the image are illuminated by different light sources was shown. Searching for global colour transformation producing a canonical illumination is in such a scene an ill-posed task. The object was however successfully recognised, partitioned according to the colour of incident light and the illumination for each part was correctly estimated.

We conclude that if there are known objects (or classes of objects, such as human faces) in a scene, that have strong geometric features, illumination-invariant recognition is applicable. Successful recognition additionally provides means for high quality colour constancy. On the
other hand, if the objects do not have distinctive parts, recognition by colour or by texture becomes necessary. In this case, colour constancy can support recognition.
8 Conclusions

In this thesis, the problem of recognition of objects in unknown scenes was studied. A complete object recognition framework was presented, which includes extraction of repeatable image regions, extraction of local coordinate systems covariant with local affine transformations, geometrically and photometrically invariant representation of local image appearance, and an efficient organisation of the object database, which allows for fast recognition response. Experimentally it was shown that the system achieves close to real-time recognition and localisation of multiple objects, and in performance compares well with other state-of-the-art methods.

The proposed method recognises instances of specific objects in large variety of scenes. The objects are assumed to be rigid, and are required to posses distinctive surface albedo. Textureless objects are not recognised. Also the problem of categorisation was not addressed, the system does not generalise from specific objects to object categories. Instead, the stress was put on the recognition from largely different viewpoints and in different illumination conditions.

A single training image per object suffices to achieve recognition invariant to significant viewpoint and illumination changes. Object representation is learned automatically from training images, without manual intervention. Maximally Stable Extremal Regions are detected and multiple affine-covariant geometric primitives are computed. Combining these primitives, local coordinate systems are constructed and used to extract affine-invariant measurements from the images. The primitives were categorised in the text, their affine covariance theoretically proven, and computational details were given.

The recognition problem was formulated as a search for a geometrically consistent set of correspondences of regions from query and database images. The search proceeds in two steps. First, a tentative set of correspondence is selected on the basis of similarity of local invariants. In a seconds step, a subset of the tentative correspondences that satisfies a global geometric constraint is found. The confidence in the presence of an object is expressed as a function of the consistent correspondences. Since it is not required that all regions match, the approach is robust to occlusion and cluttered background. And since region-to-region correspondences are established, recognition also achieves localisation.

A new type of decision tree was proposed as a database organisation, which supports matching in time sublinear with respect to the number of objects in the database. The tree was optimised for minimal retrieval time, and localisation uncertainty of LAF detection was explicitly considered. A discrete cosine transform based descriptor of local appearance was proposed. The DCT representation is computationally and memory efficient, and in the recognition performance is on par with the standard SIFT representation.

Large-scale experiments on publicly available datasets (COIL-100, ZuBuD, FOCUS) were presented. Changes of scale and illumination conditions, out-of-plane rotation, occlusion, local anisotropic scaling, and 3D translation of the viewpoint are all present in the test problems. The recognition results were compared to other published results, the performance of our system was on the top.

Finally, we have shown that illumination insensitive recognition can solve the colour constancy problem. We have demonstrated that even under severe changes of illumination many objects are reliably recognised if relying only on geometry and on invariant representation of local colour appearance. In scenes where object recognition was achieved, global photometric transformation was estimated from locally corresponding image elements, allowing high precision colour correction.
Bibliography


Biblography


Bibliography


