JOINT UTILIZATION OF LOCAL APPEARANCE DESCRIPTORS AND SEMI-LOCAL GEOMETRY FOR MULTI-VIEW OBJECT RECOGNITION

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

BY

MEDENİ SOYSAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

MAY 2012

Approval of the thesis:

JOINT UTILIZATION OF LOCAL APPEARANCE DESCRIPTORS AND SEMI-LOCAL GEOMETRY FOR MULTI-VIEW OBJECT RECOGNITION

submitted by **MEDENİ SOYSAL** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. A. Aydın Alatan
Supervisor, **Electrical and Electronics Eng. Dept., METU** _____

Examining Committee Members:

Prof. Dr. A. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU _____

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering, METU _____

Prof. Dr. Yasemin Yardımcı Çetin
Information Systems, METU _____

Assist. Prof. Dr. Selim Aksoy
Computer Engineering, Bilkent University _____

Assist. Prof. Dr. Pınar Duygulu Şahin
Computer Engineering, Bilkent University _____

Date: _____

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

Name, Last Name: MEDENİ SOYSAL

Signature :

ABSTRACT

JOINT UTILIZATION OF LOCAL APPEARANCE DESCRIPTORS AND SEMI-LOCAL GEOMETRY FOR MULTI-VIEW OBJECT RECOGNITION

Soysal, Medeni

Ph.D., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. A. Aydın Alatan

May 2012, 195 pages

Novel methods of object recognition that form a bridge between today's local feature frameworks and previous decade's strong but deserted geometric invariance field are presented in this dissertation. The rationale behind this effort is to complement the lowered discriminative capacity of local features, by the invariant geometric descriptions. Similar to our predecessors, we first start with constrained cases and then extend the applicability of our methods to more general scenarios. Local features approach, on which our methods are established, is reviewed in three parts; namely, detectors, descriptors and the methods of object recognition that employ them. Next, a novel planar object recognition framework that lifts the requirement for exact appearance-based local feature matching is presented. This method enables matching of groups of features by utilizing both appearance information and group geometric descriptions. An under investigated area, scene logo recognition, is selected for real life application of this method. Finally, we present a novel method for three-dimensional (3D) object recognition, which utilizes well-known local features in a more efficient way without any reliance on partial or global planarity. Geometrically consistent local features, which form the crucial basis for object recognition, are identified using affine 3D geometric invariants.

The utilization of 3D geometric invariants replaces the classical 2D affine transform estimation/verification step, and provides the ability to directly verify 3D geometric consistency. The accuracy and robustness of the proposed method in highly cluttered scenes with no prior segmentation or post 3D reconstruction requirements, are presented during the experiments.

Keywords: local features, geometrical descriptors, geometrical invariants, planar object recognition, multi-view object recognition

ÖZ

ÇOK AÇILI OBJE TANIMA İÇİN YEREL GÖRSEL TANIMLAYICILARIN VE YARI-YEREL GEOMETRİNİN BİRLİKTE KULLANIMI

Soysal, Medeni
Doktora, Elektrik Elektronik Mühendisliğ Bölümü
Tez Yöneticisi : Prof. Dr. A. Aydın Alatan

Mayıs 2012, 195 sayfa

Bu tez kapsamında obje tanıma için günümüzün yaygınlaşmış yerel görsel öznitelik tabanlı altyapıları ile geçen on yılın güçlü ama arka planda kalmış geometrik değişmezlik alanı arasında köprü kuran özgün metodlar sunulmaktadır. Yerel görsel özniteliklerin sınırlı ayırıcılık potansiyelinden dolayı ortaya çıkan zayıflığı, değişmez geometrik tanımlayıcılar kullanarak tamamlamak bu çabanın arkasındaki mantığı oluşturmaktadır. Daha önceki çalışmalarda olduğu gibi bu çalışmada da önce daha kontrollü durumlarla başlanmış ve daha sonra metodların uygulanabilirliği daha az kontrollü durumlara genişletilmiştir. İlk aşamada önerilen metodların dayandığı yerel görsel öznitelikler tabanlı yaklaşım, algılayıcılar, tanımlayıcılar ve bunları obje tanıma için kullanan literatürdeki metodlar olmak üzere üç farklı kısımda incelenmiştir. Daha sonra, yerel görsel özelliklere göre kesin eşleştirmeler yapma gerekliliğini ortadan kaldıran ve böylece özniteliklerin gruplar halinde ve geometrik tanımlayıcılar kullanarak eşlenmesine izin veren özgün bir düzlemsel obje tanıma metodu sunulmuştur. Bu metodun gerçek hayata uygulaması olarak bu güne kadar şaşırtıcı derecede az incelenmiş bir alan olan sahne logo tesbiti seçilmiştir. Son olarak, yerel öznitelikleri daha verimli şekilde kullanan ve düzlemsellik varsayımına ihtiyaç duymayan bir üç boyutlu (3B) obje tanıma metodu

sunulmuştur. Obje tanıma için kritik olan geometrik olarak uyumlu yerel öznitelik grupları, bu kapsamda, 3B ilgin geometrik değişmezler kullanılarak belirlenmektedir. 3B ilgin geometrik değişmezlerin kullanımı, klasik 2B ilgin dönüşüm kestirimi/doğrulaması adımının yerini alarak, 3B geometrik uyumluluğun direkt olarak denetlenbilmesine imkan sağlamaktadır. Önerilen metodun, herhangi bir ön bölütleme ya da ek 3B oluşturma adımına ihtiytaç duymadan, içerek açısından kalabalık sahnelerdeki başarılı performansı deneylerle ortaya konulmuştur.

Anahtar Kelimeler: yerel öznitelikler, geometrik tanımlayıcılar, geometrik değişmezler, düzlemsel nesne tanıma, çok açılı nesne tanıma

To my father, whose absence will always be felt

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my supervisor Prof. Dr. A. Aydın Alatan for his invaluable guidance and encouragement throughout my graduate studies. I am also grateful to all the members of my Ph. D. thesis committee for their valuable criticism and suggestions.

I would also like to thank all former and current members of TÜBİTAK UZAY Video and Audio Processing Group. They have been the essential ingredient of the creative and collaborative environment, which has proved fruitful during my studies. Special thanks are due to Dr. Ersin Esen, Tuğrul Kağan Ateş, Devrim Tipi, Ahmet Saracoğlu, Banu Oskay Acar and Dr. Özlem Birgül for their valuable help at times when my burden became too heavy for me.

I would also like to thank my former colleagues and eternal friends, Özgür Deniz Önür and Yağız Yaşaroğlu for the brainstorming sessions and spare time activities, which have always been a bliss with them.

Last but not the least, everlasting gratitude is due to my patient and supportive family members, my beloved wife Gökçen, my devoted mother Ayşe and my faithful brother Reha. They have always encouraged me to pursue success, and never left me walking alone in this long, harsh but instructive journey.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Recognition of objects has been the focus of computer vision systems for almost half a century. However, despite decades of intensive research, real life object recognition systems still has many constraints for successful recognition. At the heart of this fact lies the difficulty of adapting the optimization boundary between discriminative power and invariance against a broad spectrum of real life attacks that effect the appearance of objects. Today, utilization of local features and appearance methods, is the widespread approach for this problem. The solution, however, lies in the integration of local features with geometric constraints [1]. Geometry has played a central role within object recognition systems for a long time and still has much to offer in the context of appearance based recognition system for a few important reasons. These reasons can be listed as follows [1]:

- Invariance to viewpoint
- Invariance to illumination
- Solid theoretical framework

In order to realize the place of geometric invariants in the literature, and the evolution from formal geometry and prior models towards the use of today's appearance based statistical learning methods, it is necessary to review the research conducted on geometric recognition over the past decades.

Pioneering object recognition systems were focused on 2D pattern classification applications, such as character recognition, fingerprint analysis and microscopic cell classification. In the context of these research, geometric descriptions were used consistently, although the classi-

fication schemes vary from statistical pattern recognizers to classifiers that utilize parametric learning [2].

During those days, when geometry was a popular tool for solving recognition problems, a new approach based on the definition of the real world in terms of simplified geometrical primitives has arisen. The aim was to solve the recognition problems theoretically in a constrained world, before attacking to more realistic problems. Such a simplified environment is composed of polyhedral shapes living in a uniform environment which forms their background. Despite the simplicity that is artificially induced to this world, which is called *the blocks world*, and the geometrical convenience allowing many useful assumptions, the problem of recognizing general polyhedral shapes were not completely solved. Even so, these efforts lead to some successful applications as well as a foundation for modern geometrical systems [3, 4, 5, 6].

The blocks world representation was followed by an approach that aims to model the real world objects more closely by lifting some constraints. This representation was the generalized cylinders [7]. In this representation, curved objects were expressed in terms of a variable radius circular cross section fitted to their main axes as well as their extremities. In its evolved versions, this method's applicability was extended from simple curved shapes, such as hammer to more complex shapes, like teapot [8] and submarine [9]. These extended systems have found application in various U.S. military projects of under the names ACRONYM, SCORPIUS, SUCCESSOR and RADIUS [1, 9, 10].

The common ground in these two prominent recognition systems and their contemporaries has been the heavy reliance on 3D structure of the objects for their detection and recognition. It was assumed that knowing this structure enables a recognition system to handle the appearance modifying effects, such as viewpoint changes. This class of approaches, which embrace the idea that outlines and intensity boundaries of objects can be recovered without the need for deeper understanding of reflectance and image intensity formation, is called *object-centred representation* [1].

An alternative to the object-centered representation was first proposed by [11], in which a network of 2D views of a polyhedral shape was constructed. Almost a decade later, a complementary method, developed around the idea of pre-processing 2D views of an object in order to link common parts and form an efficient recognition plan was proposed [12]. This approach, and the underlying graph of linked object views, is called an *Aspect Graph* (Figure

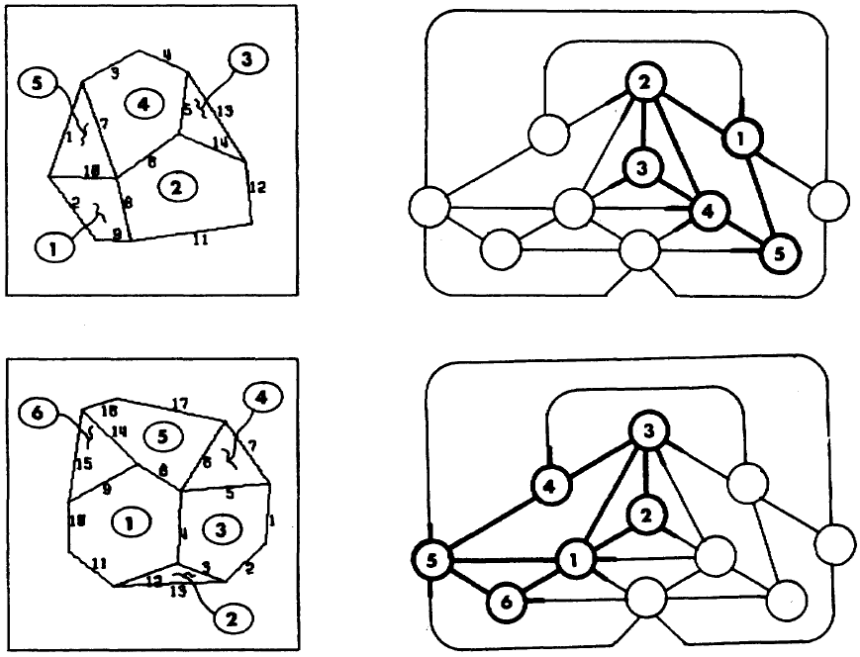


Figure 1.1: A polyhedral solid viewed from two viewpoints. Upper row: First View, Lower row: Second View, Left column: Segmented faces, Right column: Corresponding face adjacency graphs (Aspect Graphs) [14].

1.1). Object views that are adjacent to each other in terms of viewing directions, but differ significantly are the nodes of this graph. The Aspect Graph methods, however, had critical dependencies on two factors: viewpoint invariant robust segmentation and feasible connectivity complexity of segmented parts. Unfortunately, analysis of complexity showed that in the extreme conditions, the computational complexity of aspect graphs can reach n^9 for polyhedral objects with n faces and d^{18} for algebraic curved surfaces where curve degree is d [59,60]. This combinatorial explosion problem is best illustrated by a golf ball example [13], where identical groups of parts renders the model generation infeasible (Figure 1.2).

Systems of the early geometric period, independent of object-centred or viewer centred representation selection, suffered from the same presumption: Boundary descriptions defining repeatable parts of objects could be formed reliably from 2D images. This presumption was bound to collapse in typical real life cases of low contrast boundaries, background clutter with many edges and occlusion by objects with significant texture. Unveiling of this long ignored fact triggered the development of a new class of systems, which operate under the assumption that no perfect or even reliable segmentation is possible [15, 16, 17, 14]. Embracing the idea of Goad [12] that the search for features can be planned beforehand, and accepting the limited performance of segmentation lead to an efficient path of object recognition research. Lowe's

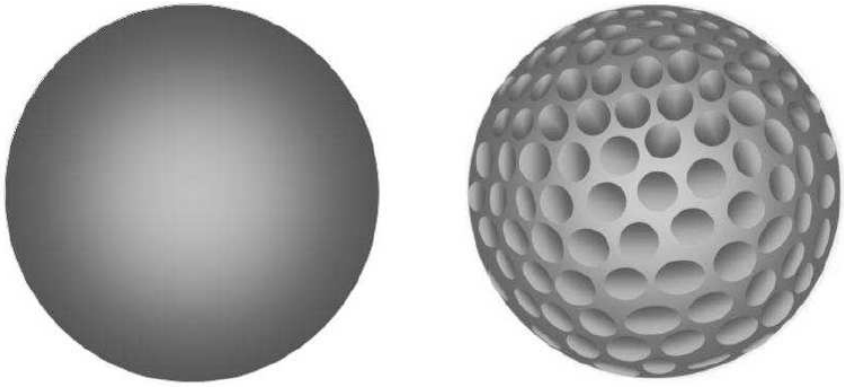
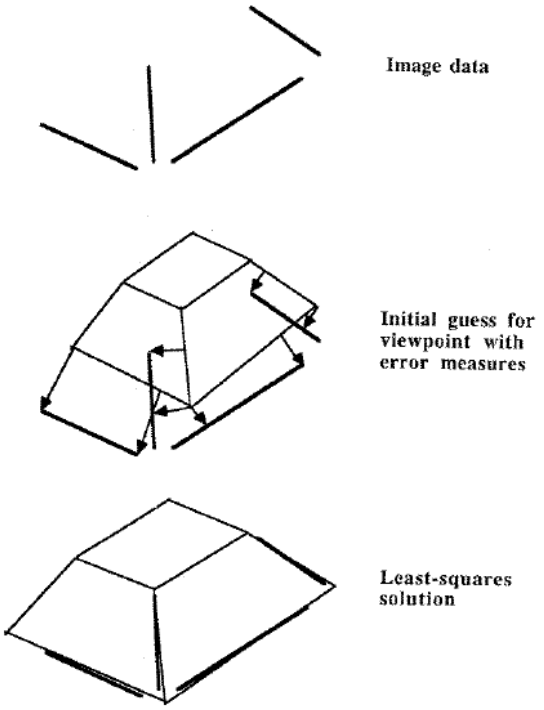


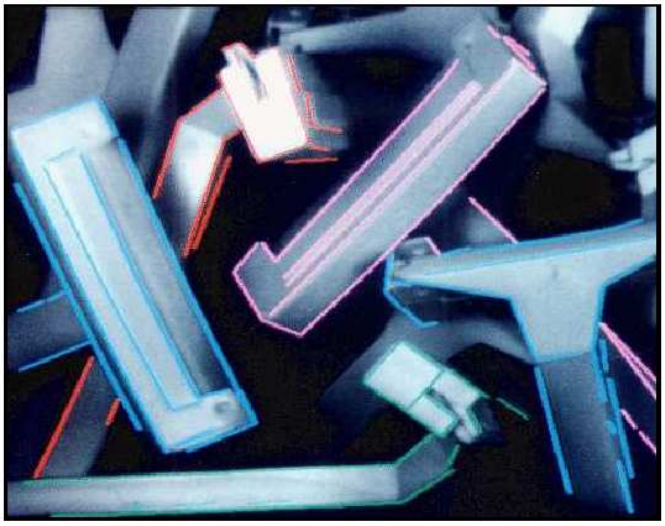
Figure 1.2: Combinatorial explosion problem. A golf ball observed at different detail or scale levels. Recursive, indiscriminative connectivity leading to an infinite aspect graph [1].

SCERPO [15], presented the first successful implementation of a recognition system based on independent local features that can be segmented reliably(edges and their intersections). In contrast to the previous systems, in this system feature perceptual grouping and linking is not required in advance of the recognition stage. Instead, model constraints were imposed on the image during the recognition step, in order to constrain the viewpoint [18, 16, 17, 14]. An illustration of recognition steps in [18] is given in Figure 1.3.

Systems that rely on minimal feature organization and strong model constraints first used based on the constrained problem of recognizing 2D planar shapes. Their rationale was as follows: Before attacking to the harder and realistic problem of 3D object recognition, it is an obvious preliminary step to solve the 2D planar object recognition problem more robustly. One of these systems exploited an interpretation tree for matching features based on their orientation information [14] (Figure 1.4). Another 2D approach, called Geometric Hashing [19], proposed a solution based on hashing of 2D point coordinates in a basis formed by three points. This approach is highly dependent on the discovery and initial matching of basis triplet for a successful recognition. In order to achieve this goal, during the pre-processing step, coordinates of all model points are repeatedly calculated according to an exhaustive set of possible triplets. This way, repeated calculation during the testing phase is prevented. This approach was later extended to recognition of 3D objects [20]. In a parallel strand of research, Ikeuchi and Kanade [21] proposed a formal definition of a recognition planning system utilizing 3D orientation constraints based on photometric stereo. Their recognition system, which takes into account both the shape and its self-occlusion model, enumerates various aspects of a recognition problem from both object and detector side exhaustively and



(a)



(b)

Figure 1.3: Steps of a model recognition algorithm based on oriented edge segments [18].

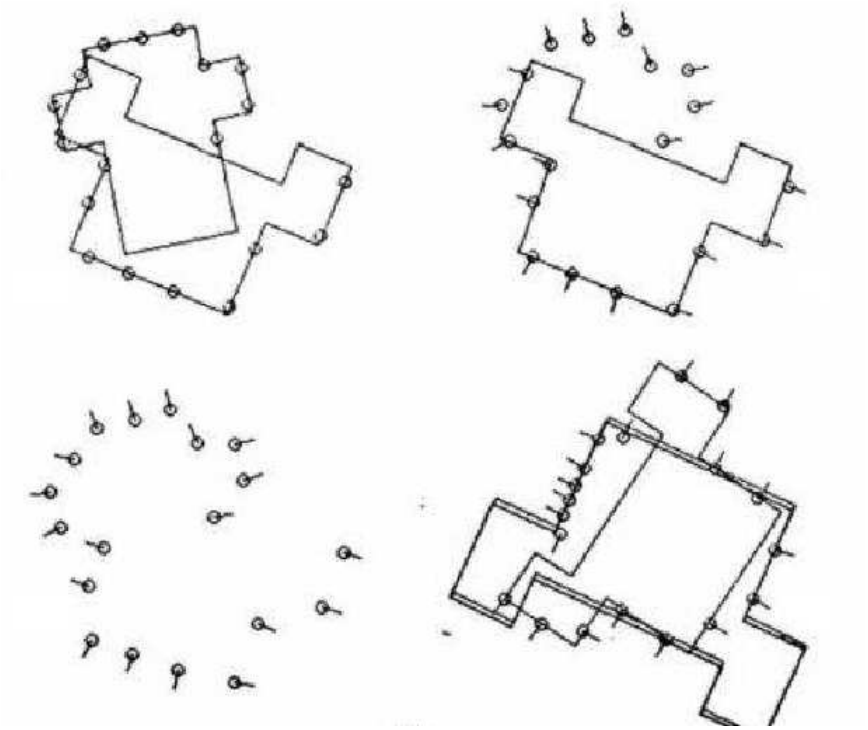
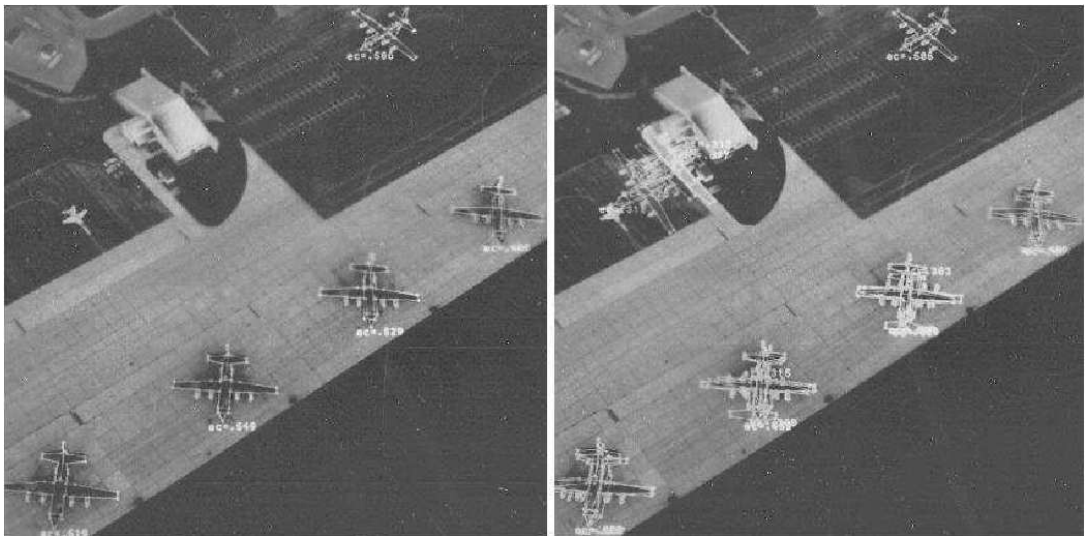


Figure 1.4: Another recognition algorithm based on oriented edge segments [1].

presents it as computer program. Such a presentation comprises an exhaustive list of sensors in object recognition, ranging from 2D edge detectors to 3D range sensors.

As soon as 2D object recognition techniques has evolved into a more mature state and after their error statistics were extensively studied [22], 3D object recognition became the focus of attention once more. This time, however, constraints were on image formation. Instead of a more general and realistic full-perspective model, projection from 3D to 2D is assumed to be behaving according to the affine projection models. One of a series of research under this category is based on edge features and their relative positioning [23]. In this work, object recognition problem is posed as an alignment problem. Alignment is performed via point triplets that are formed exhaustively resulting in a complexity of mf^3, for a comparison involving m model triplets and f feature points in a 2D image. Another research converted the model matching problem to a hypothesis testing problem, in which tests were performed by pose clustering [24]. Pose clustering is performed in a way similar to the Hough Transform [25, 26], but this time in the affine transformation parameter space. This method was then applied extensively to airplane detection problem from aerial images and reported impressively high performance results on a realistic dataset [27]. These methods, which are appropriately called *viewpoint consistency approaches* in the literature, have reported suc-



(a) (b)

Figure 1.5: Recognition results of a viewpoint consistency based method [1]

cessful results on non-complex backgrounds (Figure 1.5. This lead to their widespread use, even though their reliance on a detailed 3D model for object recognition limited their usage [28, 29, 30].

Since the beginning of 1990s, interest in geometric invariance, which has its roots in the research around ”The Blocks World” of 1960s, has started to increase again. Properties of objects that are invariant against viewpoint changes were revisited by the hope of exploring new ways to utilize geometric constraints. These developments bring constraints, such as collinear segment lengths and cross ratios that are invariant under affine and perspective viewing conditions respectively, into focus. Projective geometry, which was initially relevant to the area of computer graphics, became the main focus of the computer vision research [1]. Again, the problem was to be tackled starting from some simpler to more complex situations, and planar shapes were attacked first [31, 32, 33, 34]. This decision was mainly due to the completeness of the mathematical background explaining the theoretical foundations of affine and perspective projection phenomenon [35]. In the meanwhile, developing and extending the geometrical theory to 3D objects was also a common hope. Extending the utilization of geometric invariants to 3D objects faced two main problems during this period. The first was the irrefutable proof by Weiss that no viewpoint invariants exist for general 3D shapes [36]. The second barrier in front of the usage of geometric invariants was the grouping problem which had been the curse of the early geometric methods. For 3D case, this problem is actually much more serious due to the number of points required as the support for the invariants. Despite

these fact, methods targeting a number of constrained classes of objects, such as surfaces of rotation and polyhedra, were successfully developed [32, 37]. These methods, however, avoid dealing with the problem of grouping in order to concentrate on discovering new invariants that can be integrated to representations used in recognition systems. Later, the domain of 3D objects and image formation conditions that can be recognized using geometric invariants was extended by [38], although grouping problem was still not solved. Inevitably, the progress in geometric invariant based object recognition was hampered by the lack of invariantly repeatable features that can be detected in numbers that are high enough for invariance computation and have proper grouping constraints.

Successful research on modeling the appearance variations in images [39, 40] draw focus to this field. Geometric invariance methods were quickly abandoned in favour of appearance based methods. As the intensity appearance research deepened, methods for solving problems, such as invariance against illumination changes, were developed [41]. However, the real revolution in appearance-based methods were yet to come. In 1994, development of scale-space theory triggered this revolution that lead to a brand new approach to appearance definition [42]. This new paradigm was called ”local features” and it is still dominating today’s research. These local interest regions that constitute basis for the selection of important details in an image. Their effectiveness is evaluated according to their repeatability performances under various image formation attacks, such as scale change and viewpoint change [43, 44, 45]. Additionally, development of local detectors quickly lead to the development of local region descriptors. These region descriptors are evaluated according to two metrics: Distinctiveness and invariance [46, 45].

1.1 Scope of Thesis

In this dissertation, novel methods that combine geometric invariants and invariant local descriptions for instance-level object recognition are investigated. The problem is first addressed in a subset of the problem domain, namely planar object recognition. A novel planar object recognition method that utilizes *barycentric coordinates* in tandem with local invariant feature descriptions is iteratively developed during the initial phase of this research. Building upon the experience and insight gained during this initial phase, the problem domain is extended to general 3D object recognition, which requires a higher level of invariance against photometric

and geometric transformations.

1.2 Contributions

In this dissertation, we present novel methods of object recognition that form a bridge between today's local features framework and previous decade's mature geometric invariance field. The rationale behind this effort is to complement the lowered discriminative capacity of local features, by the invariant geometric descriptions. In Chapter 4, as the result of the first phase of the conducted research, a novel specific object recognition framework that delays exact local feature matching, enabling matching groups of features at the recognition step utilizing both appearance information and group geometric descriptors is presented. Experimental results proved the success of the proposed method in realistic datasets. Following the insight gained from the experimental results of this phase, and utilizing the geometric invariants of 3D transformations, a novel 3D object recognition method is presented in Chapter 5. Proposed 3D object recognition method performs geometric verification of local appearance-based ambiguous matches by using constraining relations between 3D and 2D invariants. Experimental results that are obtained on a well known object recognition dataset revealed the robustness of the algorithm against common attacks in this problem domain.

1.3 Outline of Thesis

The rationale behind using local feature descriptions and geometric invariants is discussed in Chapter 2. In this chapter, local features approach is reviewed in three subsections, namely, detectors (Section 2.1), descriptors (Section 2.2) and the methods of object recognition that employ them (Section 2.3). In addition, weaknesses and limitations of these methods will be reviewed briefly. Chapter 3 introduces geometric invariants (Section 3.3) along with the transformation models (Section 3.1.2) and approximations (Section 3.2) that form the appropriate basis for their derivation. Chapter 4 revisits the shortcomings of local feature-based methods and present a novel specific object recognition framework that delays exact local feature matching, enabling matching groups of features at the recognition step utilizing both appearance information and group geometric descriptors. Along with three extensions of this framework, in Chapter 4, the problem of scene logo recognition is selected for real life appli-

cation of our recognition method (Section 4.2.3). This time, considering the harsh appearance variations in real life scenarios, the recognition framework presented in Section 4.1 is adapted to work with vector quantized appearance descriptions. In the light of the simulation results, a new method that aims to solve the famous ”grouping” problem is proposed in Section 4.3. This new method is applied as a plug-in within the context of the basic algorithm and the modified algorithm is tested on an extended data set (Section 4.3.3). Chapter 5 presents generalization of geometric invariant-based approach from planar objects to 3D objects. In the presented approach, geometrically consistent local feature groups, which form the crucial basis for object recognition, are identified by exploiting the relations between affine 3D and 2D geometric invariants. The main contribution of the proposed approach lies in this ability of incorporating highly discriminative affine 3D information much earlier in the process of matching in comparison with its counterparts. The performance of the method is evaluated in highly cluttered scenes, without any prior segmentation or post 3D reconstruction requirements. These evaluations provided strong clues that suggest the promise of the proposed method. Finally, Chapter 6 closes the dissertation with a summary of our contributions and discussion of possible extensions and future research directions. The work described in this dissertation has been previously published in [47, 48, 49] and submitted for publication to [50, 51, 52].

CHAPTER 2

LOCAL FEATURES APPROACH

Local features can be defined as image patterns that differ from their neighborhood. This difference may be the result of a change of a single image property (e.g. intensity) or multiple properties simultaneously (e.g. color, texture). These local features can be points, edges or small image patches (blobs). They are typically described in terms of the properties of the region around them. The low-level descriptions extracted from each of these regions along with the spatial properties of the region (location, scale, orientation, shape, etc.) are then used for a broad spectrum of systems and applications [44].

Alternatives to local features can be broadly categorized as global features, image segments or sampled features. Global features have been used in image retrieval field for a long time. Many global features are defined in order to represent the image content in terms of color and its variations that are frequently called *texture*. These features perform quite successfully in applications where the overall composition of the image is of utmost importance. However, global features while considering the image as a whole, uses the background and foreground information without distinguishing them. Despite this fact, global features have been used in object recognition field and obtained surprisingly well results. These results actually drive the appearance based object recognition trend, leaving the previous popular trend of purely geometry based approaches [39]. However, due to major problems of image clutter and occlusions, applicability of these methods are limited to cases with bland backgrounds [53, 54, 55, 56].

Segmentation can be used to solve the problems experienced with global features. This term can be defined as dividing the image into a limited number regions depending on a consistency constraint. The goal is to perform this operation so as to obtain segments corresponding to the semantic parts or single objects. Despite some advanced applications that are found in

the literature, such as the *Blobworld* [57] that employs both color and texture consistency to divide the image to semantically meaningful segments, image segmentation is still a very challenging and complex task. This difficulty leads to the common opinion that segmentation is part of a chicken-egg problem, and in some situations the end result of high-level image understanding becomes much practical than a complete segmentation.

In order to avoid segmentation, but still stay less prone to the clutter and occlusion effects that are experienced in the global case, densely sampled features can be used. This approach depends on exhaustive sampling on different parts of the image at many scales and orientations. Descriptors extracted from each of these parts are then compared. This approach is called as the *sliding window* in the literature, and is very popular in face detection and pedestrian detection applications after the popular work of Viola and Jones [58]. These methods can be considered as brute force, since they analyze each and every sub window in the image. Such an approach brings the requirement that only extremely efficient implementations can be used in real life situations. As an alternative to sliding window approaches, fixed grid approach is introduced. In this approach, image is sampled sparsely from a fixed grid of locations. They are robust against occlusions and scale changes up to some extent, rendering them useful in scene classification or texture recognition applications [59]. On the other hand, they do not possess the required localization power for being useful in applications that require precise location information. Another similar approach is using *random sample patches*, which is actually a random generated list of location, scale and shape triplets. Similar to the fixed grid methods, these achieved good scene classification results due to their dense coverage of the images [60, 61]. These random patches, however, also lack the proper location information that is repeatable and therefore essential to more complex applications.

Local features raise among these approaches with some properties that render them well suited to more complex recognition applications, such as instance level object recognition. Their significance is twofold: First, they provide a robust way to represent the images in terms of parts without a requirement for an explicit segmentation. Second, they provide a computationally feasible number of well localized and individually identifiable anchor points. Although these features do not necessarily correspond to human friendly semantics with a plausible verbal interpretation, they provide reliable cues whose location and other spatial descriptions remain stable under a wide range of attacks. Obviously, local features that correspond directly to semantically meaningful object parts would be preferable ideally. But then, this goal would

be similar to the segmentation problem and require high-level interpretation of the image content. It is important to define a good local feature, which can be obtained directly from the intensity patterns without the need for any high-level information, in an objective way. A preferable local feature can be identified by the following properties:

- Repeatability
- Distinctiveness
- Locality
- Quantity
- Accuracy
- Efficiency

Repeatability is the observability rate of features detected at similar locations at images with similar content. It is a prerequisite for any application that use local features. Invariance to large deformations or robustness to small deformations is required in order to obtain an adequate number of repeatable features.

Detected features are most useful, if they are located on informative local patterns. This is called as the distinctiveness property. Since local features have a small supporting region, their characteristic properties are limited. Therefore, the regions that stand out are highly valuable.

The localization of the detected features should be good enough to allow for extraction of descriptions that are unaffected by occlusion. Besides, as the size of the interest region increases it becomes hard to stay robust against geometric and photometric deformations, resulting in the loss of repeatability.

Although the optimum number of required local features is highly dependent on the application, it is necessary to be detectable even on small objects. The number of features detected should reflect the density of the image information.

The features should be detected at almost the same location and scale (and sometimes shape) in order to offer possible correspondences.

Lastly, they should not be computationally too complex in order not to hamper time-critical applications.

In this section, we will first provide a summary of a selected subset of local feature detection methods which are particularly important in the object recognition field. In the context of this summary, favorable properties, such as repeatability under a set of transformations will be stressed out whenever possible. Next, we will analyze low-level feature descriptions that are used to represent the appearance characteristics of local features, and to compare them for finding correspondences. Again, invariance properties that are important from the recognition point of view will be examined briefly. Lastly, we will summarize methods in the literature that utilize local features in the context of object recognition.

2.1 Local Feature Detection

The first step of an approach based on local features is detection. Local feature detection research dates back to around fifty years from today. Many methods were proposed in this period for extracting repeatable, robust and precise local features automatically from images. The beginning of the research on primitive local structures, which are considered as interesting by the human visual system, was marked by a psychological analysis [62]. In this analysis, corners and junctions stand out as important cues for visual recognition. This result was later generalized as contours with high curvature, extending the target local structures to intersections and junctions that have a high ratio of unit tangent vector change per arc length.

On the other hand, there has been another strand of research which concentrated directly on image intensities and high variances in them as indicated by derivative calculations. Harris [63], Hessian [64] and Smallest Univalue Segment Assimilating Nucleus (SUSAN) [65] corner detectors are a result of this strand of research.

Another research with a significantly different motivation was based on modeling the human visual system by discovering biologically plausible methods for feature extraction. The pioneer of this type of approach was Marr through his influential work [66] where he provides a deeper understanding of biological visual perception. Laplacian of Gaussian (LoG) and Gabor filter response based detectors are famous representatives of this approach.

In parallel, analytical study of corner detection, which can be represented under model based local feature detection, was conducted. This research resulted in detectors that function either by fitting masks [67] to match the underlying intensity structure or by fitting a parametrized model using novel techniques, like Hough Transform [68].

Introduction of *Scale-Space Theory* [42] provided a theoretical foundation for the detection of scale for local features that are otherwise defined only by their location. Following the development of this theory, many local feature detectors were modified to perform automatic scale selection [43].

Even segmentation techniques have also been applied to local feature extraction problem. Despite the well known fact that optimal segmentation is intractable in general, several systems for segmentation-based local features were developed for recognition purposes. *Maximally Stable Extremal Regions (MSER)* [69] that utilize a watershed-like segmentation algorithm and *Intensity based Regions (IBR)* [70] that are formed by detecting the boundaries of the region around a local intensity by sweeping the neighborhood with a 1-D intensity sampling ray are two prominent examples of segmentation based techniques.

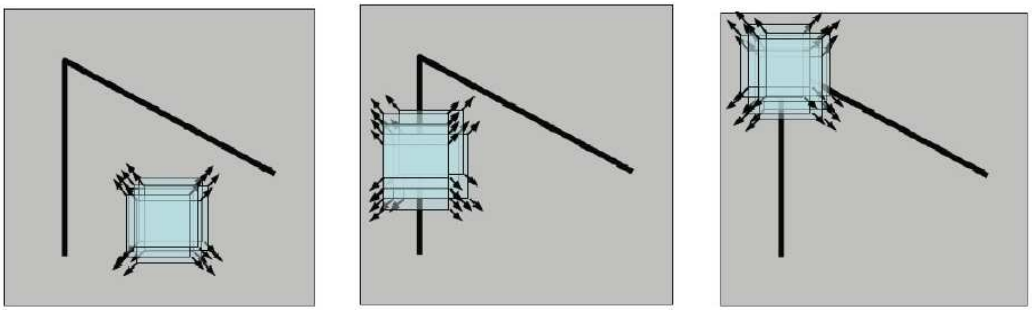
In this section, we will provide a brief summary of the local feature detection research via presenting its most prominent products, which have significant and widespread influence on the object recognition field. These methods can be broadly presented in three main categories which are designed to group the detectors based on the types of local features detected by them, namely corners/edges, blobs and regions [71].

2.1.1 Corner / Edge Detectors

Harris Corner Detector [63] is one of the most reliable local feature detectors [44]. It is based on the second moment matrix in Equation (2.1), which is formed of the first degree terms in the Taylor series expansion of the image intensity $I(x, y)$

$$M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2.1}$$

Harris detector operation essentially corresponds to measuring the local changes of an image via shifting patches around the position under consideration by a small amount in different



(a) (b) (c)

Figure 2.1: Autocorrelation matrix responses around different types of points. (a) Both eigenvalues are small, the windowed image region is of approximately constant intensity (b) One eigenvalue is high and the other is low, the windowed region is considered to contain an edge (c) Both of the eigenvalues of the auto-correlation function are high, variance high in all directions.

directions. The relation between the two eigenvalues of the matrix in Equation (2.1), which is computed during this operation, categorizes the image structure under consideration as belonging to one of three classes given in Figure 2.1. Harris Cornerness Measure, which is defined as $|M| - \alpha \cdot \boldsymbol{Tr}(M)$, discriminates these structures by selecting the third one, in which local neighborhood under consideration is highly variant in any possible direction. This metric is considered as an accurate clue for marking the region under consideration to contain a corner.

Harris corners are highly repeatable under photometric transformation affecting the appearance of local patches. Since only the derivatives are used during the computation of the cornerness measure, it is invariant even under harsh affine intensity changes. On the other hand, Harris corners are also robust against geometric translation due to the autocorrelation function approximation it adheres. In addition, utilization of eigenvector magnitudes, renders the cornerness output invariant under geometric rotation operation (component of the Euclidean Transformation that is illustrated in Figure 3.3). Hence, in its basic form, one can say that Harris corner detector results are quite repeatable. Geometric scale and affine changes, however, are beyond the scope of the invariance scheme of the basic Harris detector and needs special treatment. A visual list of the transformations mentioned above are given in Figure 3.3.

Harris corner detector only gives location information for the detected local features. In order to deal with scale changes, an automatic scale selection method proposed by Lindeberg can be adopted [43]. This method is appropriately called *Harris-Laplace*, since it involves the search

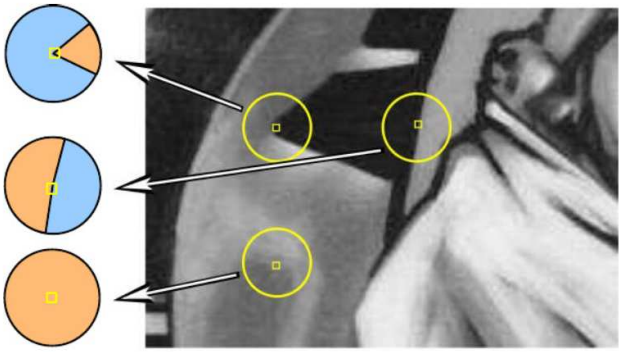


Figure 2.2: Edges detected by SUSAN detector [71]. Red areas are pixels that are similar to the nucleus and blue ones are different.

for a characteristic scale of the local structure using the Laplacian function. Characteristic scale is identified by the scale-space extremum of the scale adapted Laplacian function, which is defined as [43]:

$$\nabla^2_{norm} I(x, y; \sigma^2) = \sigma^2 (I_{xx} + I_{yy}) \tag{2.2}$$

where $I(x, y; \sigma^2)$ represents the image convolved with a 2D Gaussian function of standart deviation σ. Local features found using this detector are repeatable under similarity transformations.

Affine transformation is the common type of transformation for small planar patches viewed from a distance much larger than the size of the patch. Since Harris-Laplace regions fall into this category, it is an important property to be invariant under this type of transformation. *Harris-Affine* detector achieves exactly this goal through local affine adaptation [44]. This adaptation is achieved on initial points extracted along with their characteristic scales. Elliptical shape of the region under consideration is estimated iteratively using the eigenvalues of the second moment matrix given in Equation (2.1).

The *Smallest Univalue Value Segment Assimilating Nucleus (SUSAN)* detector uses a morphological approach to detect corners [65]. Basically, on each pixel in the image, a fixed radius circular operator is applied. This operator compares each pixel in the radius with the center pixel which is called *nucleus* in terms of intensity. SUSAN corner detector finds the corner locations where the ratio of pixels similar to nucleus to the pixels different from it drops below a certain threshold (Figure 2.2). This detector is also used for edge detection and noise suppression in the literature. Recently, new low complexity techniques, such as FAST [72] and ORB [73] are also proposed, all of which have similar approaches to SUSAN.

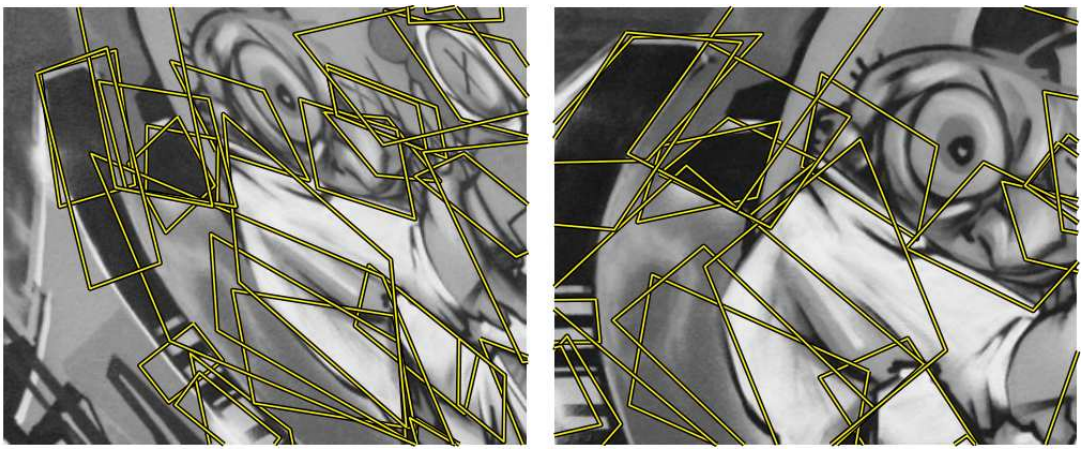


Figure 2.3: Edge-based regions detected at two different scales [71]

A prominent example of edge based techniques is the *Edge-based Regions* (EBR) that is developed by Tuytelaars and Van Gool. In this technique, edges are searched around Harris corners by the Canny edge detector. Local features are then defined as parallelogram regions whose two main axis are determined according to the speed of divergence of edge from the corner point (Figure 2.3).

Edge-Laplace features are obtained based on edges detected by Canny edge detector. Each of the detected edge pixels are then considered as a candidate for being a local feature and their Laplacian is computed at multiple scales. The edge pixel locations which has a distinctive Laplacian extremum in scale domain (Equation 2.2) is selected as Edge-Laplace features.

2.1.2 Blob Detectors

Similar to corners, homogeneous regions that "pop-out" are known to be detected in the pre-attentive stage of the human visual system. Detectors that imitate this sensitivity of our system are called as *blob* detectors. Blobs are defined as points or regions in the image that are either brighter or darker than their surrounding. Although there are other categories of local feature detectors, blob detectors are the first that are called as *interest point detectors* or *interest region detectors*. The rationale behind the development of blob detectors is to provide information about regions, which can not be extracted by edge or corner detectors. This information complements the information from corner/edge detectors (Section 2.1.1).

A prominent example of derivative based method, similar to the Harris corner detector (see

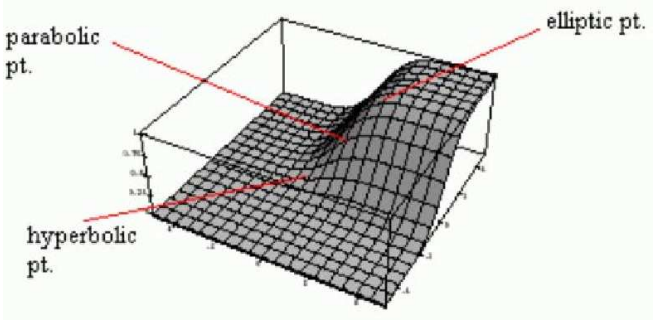


Figure 2.4: Illustration of elliptic, parabolic and hyperbolic regions.

Equation 2.1), but reacts to the presence of blobs is the *Hessian* detector, that utilizes the Hessian matrix.

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \tag{2.3}$$

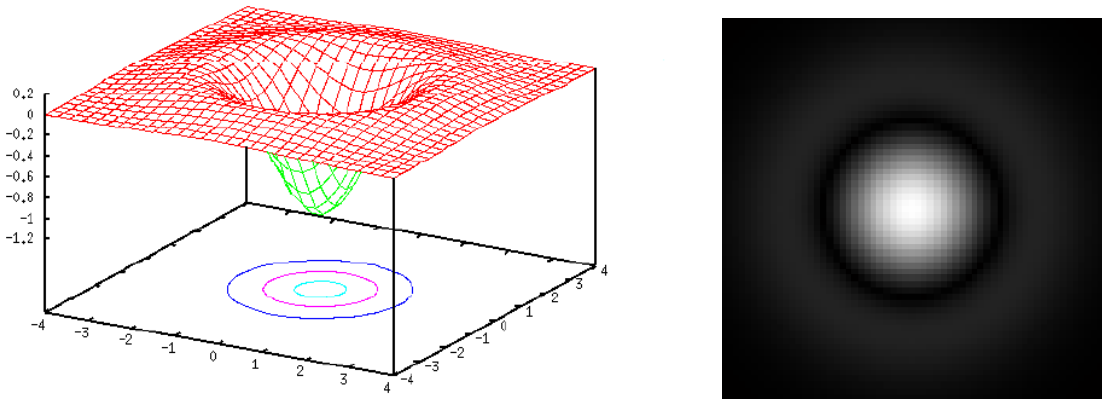
where I represents the image intensity function and the elements of the matrix are the second order derivatives of this function. Hessian detector [64] response function is defined as $I_{xx}I_{yy} - I_{xy}^2$. This function is also proportional to the product of principal curvatures given in Equation (2.4),

$$K = K_{min}K_{max} = \frac{I_{xx}I_{yy} - I_{xy}^2}{\left(1 + I_x^2 + I_y^2\right)^2} \tag{2.4}$$

and is interpreted as illustrated in Figure 2.4. Hessian detector always detects an elliptic maximum inside the corner area independently of the local image contrast. However, the distance of the maxima from the corner depends on the sharpness of the angle.

Hessian-Laplace and *Hessian-Affine* detectors are scale invariant and affine invariant extensions of the basic Hessian detector. In terms of the extension method, they are very similar to Harris-Laplace and Harris-Affine extensions that are described in Section 2.1.1. The only difference between the Harris and Hessian extensions is that Hessian extensions utilize the determinant of the Hessian matrix as the initial location detector rather than the Harris corner detector. Hessian-Laplace and Hessian-Affine detectors have been proposed as appropriate counterparts of the Harris-based viewpoint invariant detectors in the literature [74, 44].

Laplacian-of-Gaussian (LoG) is a powerful multi-scale local feature detection tool that was first used by Lindeberg for scale-invariant blob detection [43, 42]. Lindeberg represented blobs as maxima of LoG, which is defined as:



(a) (b)

Figure 2.5: Laplacian-of-Gaussian (LoG) function (a) 3D visualization of the LoG function, (b) 2D visualization of the LoG magnitude.

$$LoG = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} \tag{2.5}$$

where $G(x,y)$ is the well-known Gaussian function with variance σ^2. Maxima of the LoG magnitude are searched in both spatial (Equation 2.5) and scale domain, and therefore, Equation (2.5) should be normalized in order to perform a fair comparison among responses of different scales. This comparison is achieved by:

$$LoG = \sigma^2 \left(\frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} \right) \tag{2.6}$$

This simple operator above, which is visualized in Figure 2.5, constitute the heart of Lindeberg's famous scale-space theory, which forms the theoretical foundations of today's successful local feature detection and scale selection algorithms by showing a standard mathematical way of relating image structures between different scales.

Difference-of-Gaussian [75] is a close approximation to scale-normalized LoG, which is designed in order to speed up the detection process. Instead of the second derivative of the Gaussian function, DoG convolves the image with Gaussians at different scales and subtracts them to obtain a DoG scale-space. DoG feature points are then selected as maxima detected in the 3D scale-spatial space.

Speeded Up Robust Features (SURF) [76] is a detector optimized for speeding up the extraction process. SURF constructs a scale space similar to the DoG, but approximates the

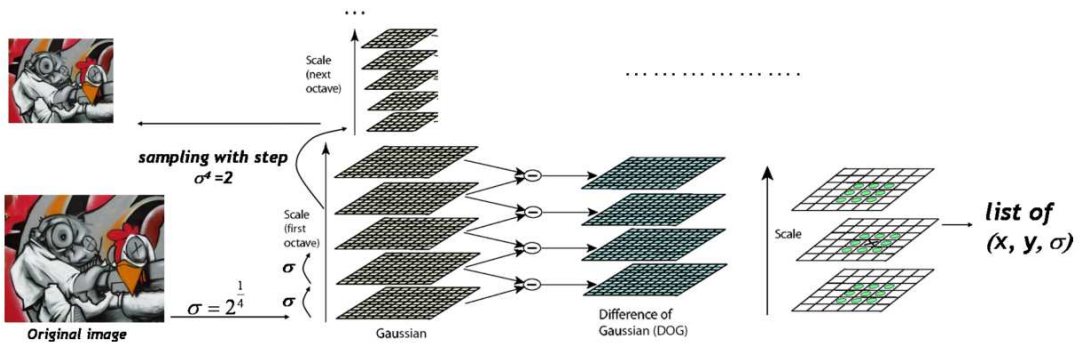


Figure 2.6: Difference-of-Gaussian (LoG) function [75]

determinant of the Hessian matrix, instead of the LoG function. Local feature locations are selected as Hessian determinant maxima in both spatial and scale domain.

Salient Regions [77] represents a novel approach to local feature detection. This detector is not based on the derivative information in the image as the other detectors. Instead, it adheres an information theoretic approach. In the context of this detector, saliency is defined as local complexity or in other words, unpredictability of the patterns in a region. This property is measured by the entropy of the probability distribution function of the intensity values within a local image region. In order to select the scale of the salient regions, magnitude of the first degree derivative of the intensity probability distribution function with respect to scale is calculated at entropy maxima. The scale-adapted saliency value for each region is then computed as the product of the entropy and this scale space derivative. ,

2.1.3 Region Detectors

Unlike the previous two categories of local feature detectors, region detectors are used to group those algorithms which are concerned with extraction of image regions. These methods initially extract complex boundaries. These complex boundaries are later approximated with a simple parametrized shape, namely ellipse, for efficiency in representation.

Intensity-based Regions (IBR) detect affine invariant regions around multi-scale intensity extremas [70, 78]. The region boundaries are then searched radially using rays emanating from this center point. This search is guided with a function that is evaluated along the ray:

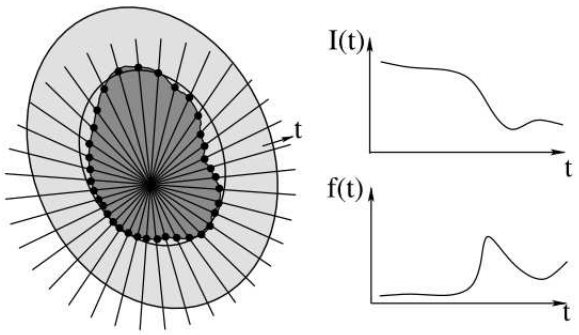


Figure 2.7: Intensity-based Regions (IBR) Extraction [44]: $I(t)$ is the image intensity along rays, $f(t)$ is the function whose extrema indicate boundary points (Equation 2.7).

$$f(t) = \frac{|I(t) - I_0|}{max\left(\frac{\int_0^t |I(t) - I_0| dt}{t}, d\right)} \tag{2.7}$$

where t is the ray parameter and $I(t)$ the image intensity at ray position t, I_0 the intensity value at the extremum, and d a small number which has been added to prevent division by zero. The points along the rays are evaluated using this function and points where an extremum occurs are selected as boundary points (Figure 2.7). The regions boundary is obtained by connecting these points, where an extremum of function $f(t)$ occurs. The resulting irregular region shape is then replaced by an ellipse fitted to this random boundary.

Maximally Stable Extremal Regions (MSER) can be defined, in simple terms, as components, which maintain their connectedness in the course of a series of thresholding operations [69]. The *extremal* term refers to another important property: All pixels that are inside a MSER have either higher or lower intensity than all the pixels on its boundary. This property also explains the determination of the region boundaries: Boundaries are selected in such a way that the area of the MSER region is the least affected from changes of the image thresholding parameter. Examples of regions obtained during this process is given in Figure 2.8

Some of the major automatic local feature detection methods that are widely used in the literature were presented in this section. According to the reported evaluation results [79, 46, 74, 71, 44, 62, 45, 80], local feature detectors which are only invariant against rotation and translation (Harris, SUSAN, EBR) has the highest localization accuracy. However, these corner/edge detectors has a lower accuracy in scale estimation (Harris-Laplace, Edge-Laplace)

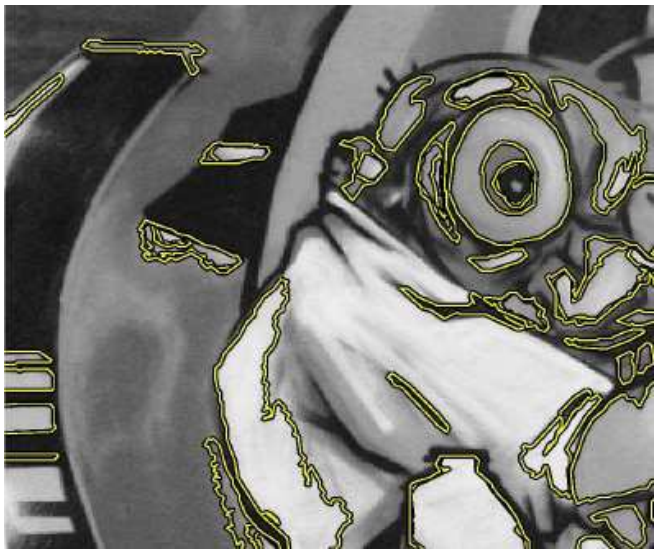


Figure 2.8: Maximally Stable Extremal Regions (MSER) Example Detections [71]

due to the multi-scale nature of corners and edges. On the other hand, blob detectors have a lower localization accuracy, but compensate this property by a higher scale estimation accuracy. Again, this is a direct consequence of the underlying blob pattern which is better localized in scale space. Region detectors, such as MSER and IBR, are known for their robustness under harsh viewpoint transformations over an angle of 30° that lead to affine distortions. Under these types of extreme geometric deformations, affine invariant detectors are mandatory. However, under more constrained transformations their scale-invariant counterparts are known to provide a better representation.

Local features are defined by both location and a related patch. Appearance descriptors that are widely used in the literature for representing these patches are introduced in the next section.

2.2 Local Feature Description and Comparison

The simplest way of describing local features that are represented by a small patch which is defined by a position and scale is of course, directly using a vector of image intensities. These vectors can then be compared by summing the squared differences (SSD) of descriptor vector elements of the compared patches:

$$SSD = \frac{1}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} (I_1(x_1+i, y_1+j) - I_2(x_2+i, y_2+j))^2 \tag{2.8}$$

where $I_1(x_1, y_1)$ and $I_2(x_2, y_2)$ represent the corresponding pixel locations in the compared patches belonging to images I_1 and I_2, respectively. Patches are of size N by N for both images. Small differences in SSD signals highly similar patches. However, the above comparison would fail to capture similarities between the projections of the same patch under basic photometric transformations, such as simple lighting changes ($I \rightarrow I + b$). This effect can be compansated by performing a normalized comparison using the mean of patch intensity values via Mean Normalized SSD (NSSD):

$$NSSD = \frac{1}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} ((I_1(x_1+i, y_1+j) - \mu_1) - (I_2(x_2+i, y_2+j) - \mu_2))^2 \tag{2.9}$$

where μ_1 and μ_2 are mean intensity values of the two patches. More complex photometric effects like affine photometric transformation ($I \rightarrow aI + b$), requires more sophisticated comparison normalized with both mean and variance of intensity values of patch pixels, corresponding to zero normalized sum of squared differences (ZNSSD):

$$ZNSSD = \frac{1}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} \left(\frac{I_1(x_1+i, y_1+j) - \mu_1}{\sigma_1} - \frac{I_2(x_2+i, y_2+j) - \mu_2}{\sigma_2} \right)^2 \tag{2.10}$$

where σ_1 and σ_2 are standard deviations of intensity values inside the two patches. These comparison methods cope well with photometric transformations, when two patches are viewed with exactly the same camera geometry. However, in real life, local features undergo a much wider spectrum of transformations that lead to a drastic change in both appearance and geometry of the local feature regions. In order to handle these transformations, various descriptors possessing invariance against different subsets of photometric and geometric transformations are proposed in the literature. This section is dedicated to a brief review of these description methods.

The most common way of robustly describing the appearance characteristics of a local feature is using a distribution based approach utilizing local histograms. The most prominent of these

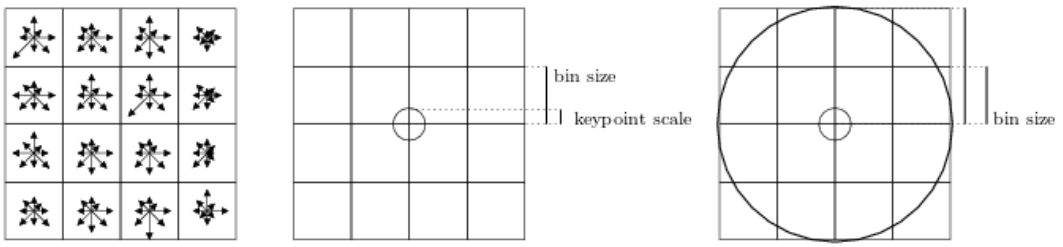


Figure 2.9: Scale Invariant Feature Transform (SIFT) (Left) Spatial Histogram of Gradients, (Middle) Dynamically Scaled Histogram Spatial Grid, (Right) Gaussian Weighted Histogram Support [81].

descriptions is the Scale Invariant Feature Transform (SIFT) [75]. It extracts a 3D spatial histogram of the image gradients. Gradients computed at each pixel is considered as a triplet defined by its spatial location and orientation. These sampled triplets are then weighted by the norm of their gradients and accumulated in the 3D histogram. During this accumulation, location is quantized into a $4x4$, grid while orientation is quantized into 8 orientation bins. In order to provide invariance against translation, scaling and rotation, the location, spatial bin size and grid orientation of the 3D histogram is determined dynamically according to local feature location, detection scale and the dominant orientation of the local patch, respectively (Figure 2.9). Its histogram-based nature, together with scale and orientation adaptation properties renders this descriptor robust against small geometric distortions and small localization errors in the local feature detection step.

Geometric Histogram [82] and Shape Context [83] are two other histogram-based description methods that are very similar to SIFT, except minor differences. They both compute equally weighted 3D histograms of gradients based on edge points inside the patch region.

Gradient Location and Orientation Histogram (GLOH), as implied by the name, is another gradient histogram based descriptor [46]. This descriptor extracts the gradient histogram from a circular grid whose size is determined by the local feature scale. In addition, the description quantizes orientation in a way slightly different than the SIFT descriptor. As a final step, original 272-dimensional descriptor is projected to a 64-dimensional space via Principal Component Analysis (PCA).

Histogram of Oriented Gradients (HoG) is among the modified variants of SIFT and is mainly used in densely sampled grid of locations. It is known to perform well on pedestrian recogni-

tion problem [84].

A rotation invariant extension of SIFT is called Rotation Invariant Feature Transform (RIFT). This descriptor is equipped with invariance against rotation, instead of its rotation covariant ancestor SIFT. As a result, the cost for dominant orientation calculation is subtracted from extraction complexity, in the expense of reduced discriminative power.

Speeded Up Robust Features (SURF), is an optimized detector-descriptor pair, whose detector is introduced in Section 2.1. SURF description is an approximation of SIFT description with significant computational efficiency improvements. These improvements are achieved by using integral images for computing derivatives and using a lower number of spatial histogram bins for gradients.

In contrast with the above approaches, spin images [85] adapted to 2D, provide a histogram representation indexed based on intensity values and relative distances of the patch pixels from the patch center [86]. The original 3D version of this descriptor is developed for recognition of distinctive points inside range (depth) data.

Local Binary Patterns (LBP) is another method describing the local appearance using a distribution, with the difference representing binary relations between intensities of neighboring pixels [87]. It computes an histogram based on binary ordering and relative comparisons of pixel intensities. In this descriptor, relations extracted from predefined pixel locations inside the local patch are encoded as a binary string. As expected, the reliability and distinctiveness is proportional with the complexity and therefore dimension of the descriptor.

As an alternative to distribution-based local feature representations, techniques that describe local frequency content can be utilized. However, capturing the small changes in frequency and orientation inside typically small local feature regions requires a large number of basis functions. Gabor filters and wavelets are prominent examples of frequency-based approaches and mostly used in the area of texture recognition [88, 89].

Local feature neighborhoods can also be represented by a set of image derivatives computed up to a predefined order. Local Jets [90], Steerable filters [91] and complex filters [92] are examples of methods in this category. These methods combine different derivative components in order to achieve invariance against various transformations. Most of these methods employ Gaussian approximations during their derivative computation step.

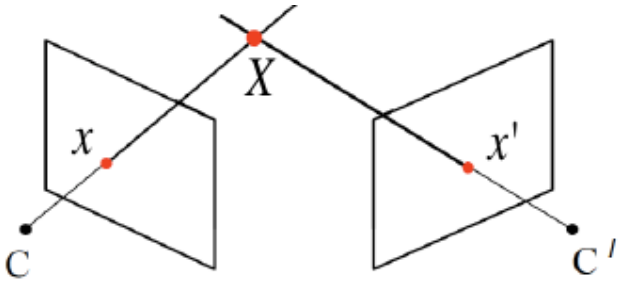


Figure 2.10: Correspondence relation between two images of a 3D scene point [95].

Generalized Moment Invariants is another local feature description method, which is significantly different from histogram-based methods [93]. This description adapts the moment invariants that are computed from binary images for shape recognition to the local feature description area by defining the central moments as:

$$M_{pq}^{a} = \int \int_{\Omega} x^{p} y^{q} [I(x,y)]^{a} dxdy \tag{2.11}$$

where $I(x,y)$ is the image intensity at position (x,y). This central moment defined by this equation is computed on local region Ω, and with an order $p+q$ and degree a.

According to the reported results [46, 94, 71, 62, 79], histogram-based descriptors that are similar to the SIFT descriptor generally perform best on object recognition problems. This performance is shown to be relatively independent from the variations in the prior local feature detection step. Utilization of the local features and their descriptions in the context of object recognition is investigated in the next section.

2.3 Local Feature Based Methods for Object Recognition

The goal of image matching ve recognition with local features is establishing correspondences between two or more images. Correspondence is used in this context to represent the relation between two image projections of the same 3D scene point (Figure 2.10).

The most basic and maybe influential application area of local feature-based correspondences is *wide-baseline matching*. Utilization of local features matching in camera calibration, 3D reconstruction, structure and motion estimation are all prominent examples belonging to this area. The most distinctive character of the problems in this area is the confident prior knowl-

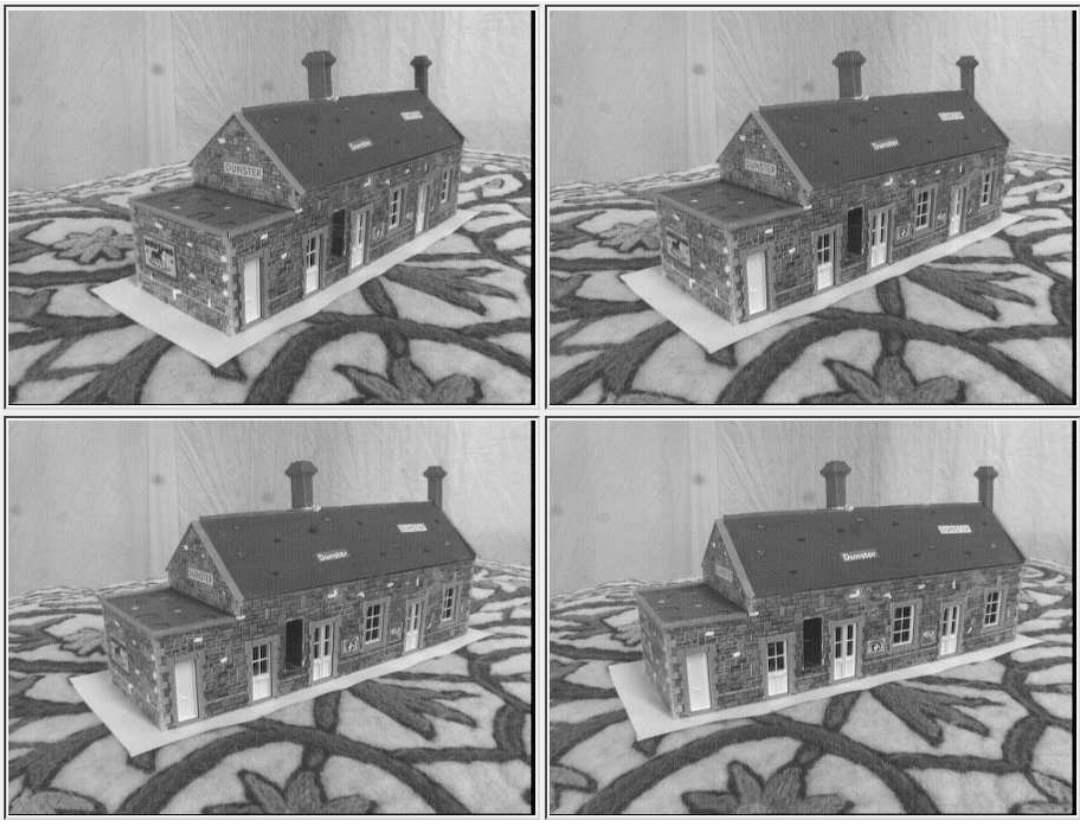


Figure 2.11: A set of typical multi-view images that are used in wide baseline matching [98]

edge that the targeted semantic content, i.e. object, coexist in both images. This in turn, means availability of true correspondences in large numbers. In addition, correspondences and objects in typical images to be matched undergo only a limited set of transformations and most of the content coexist in both of the images (Figure 2.11). Still, local feature-based matching of images under these constraints is a valuable tool for commercial applications, namely AutoStitch [96], and 3D modeling applications, such as [97].

Under the limited transformations of the wide-baseline matching area, generalization of underlying object characteristics is of limited concern. On the other hand, for the area of object class recognition or category detection, generalization of object characteristics constitutes the core of the problem. Category level generalization need to account for variances resulting from both intra class variance (Figure 2.12) and imaging condition variances (Figure 2.13) as well as occlusion, truncations and background clutter (Figure 2.14).

Representing objects with a pictorial structure diagram is adopted as an intuitive approach to the category detection problem, since it deals with all the variance issues that are mentioned

Figure 2.12: Example images of “chair” object category [99]

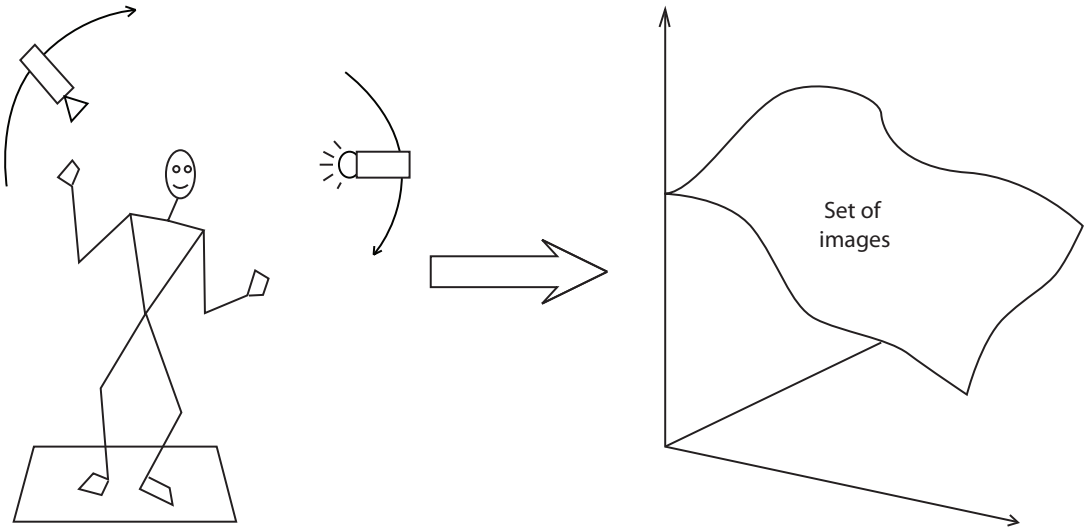


Figure 2.13: Imaging condition variations

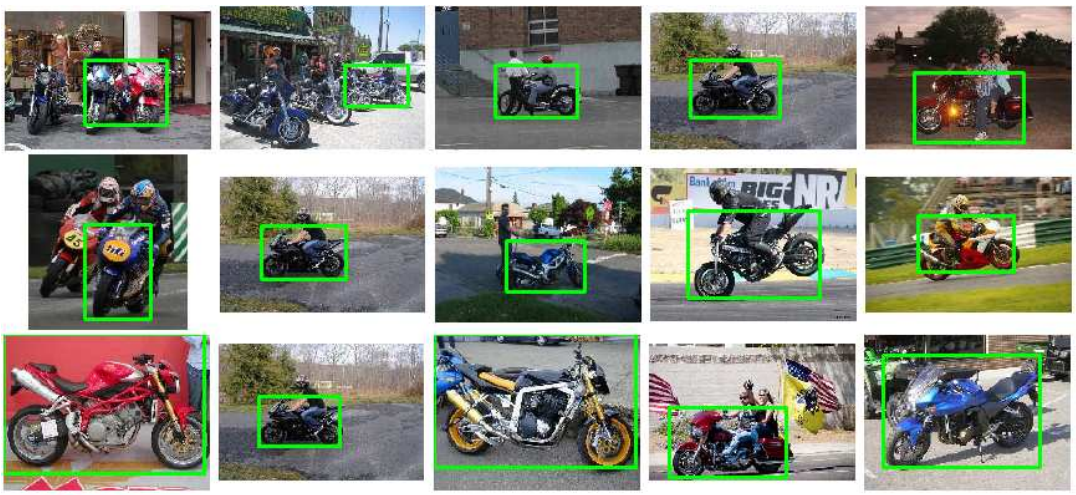


Figure 2.14: Typical images of motorcycle category with occlusion and clutter [99]

above directly. Pictorial structure approach dates back to the research conducted in 1973 by Fischler [100]. In this research, objects are modeled as having two separable components; parts and structure (Figure 2.15). The term "parts" is used to represent 2D image fragments defined by distinctive visual features. These visual features are presumably informative parts that can be robustly extracted from images containing translated, rotated and scaled versions of objects. "Structure", on the other hand, is used to represent the complementary information of collective part configuration. Approaches following the spring-like model of Fischler [100], can be grouped under two main representations. These representations, namely *fully connected shape model* and *star shape model*, differ in the way parts are related to each other in defining the structure of the object category (Figure 2.16). Constellation model and implicit shape model are famous examples of these categories.

Constellation model assumes a fully connected configuration of parts, which lead to a probabilistic model of joint spatial distribution of parts [101]. This explicit structure model is complemented with appearance based part detectors that determine position and scale of parts. The training phase utilizes images alone using EM algorithm [103] for simultaneous learning of parts and structure. Constellation model provides an abundance of elasticity that renders the models adaptable to any object category, independent of the priority order between appearance and shape. However, constellation model contains many parameters that need to be estimated and therefore, model complexity increases swiftly by the number of object parts ($(O(N^P))$ where N is the number of parameters for s single part model and P is the number of parts). As a result, this theoretically compelling model has practical issues hindering its wide

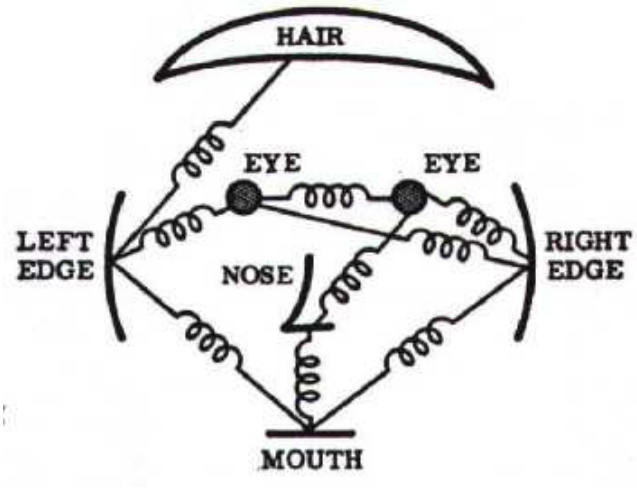


Figure 2.15: A Pictorial Structure Diagram illustration where springs represent the relations between parts [100].

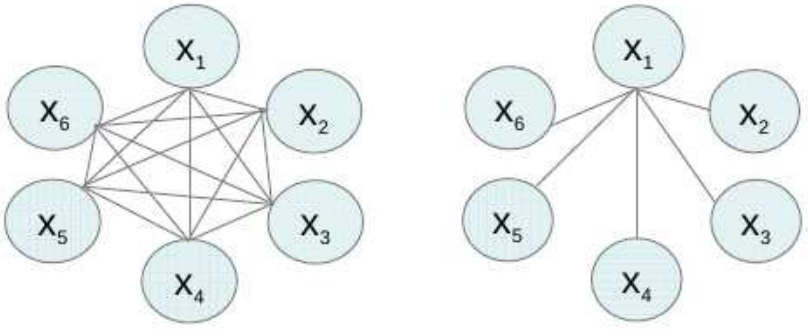


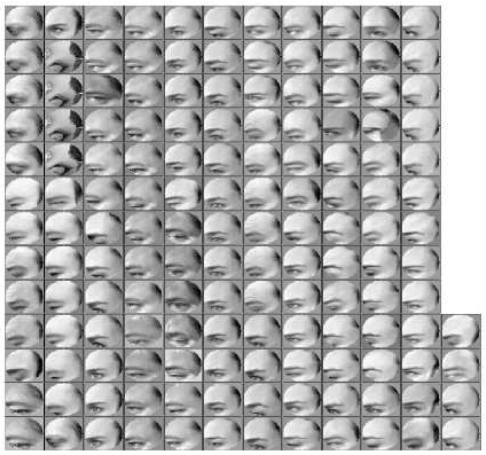
Figure 2.16: Comparison of Pictorial Models: (Left) Fully connected shape model (e.g. Constellation model [101]), (Right) Star shape model (e.g. Implicit Shape Model [102])

applicability to object categorization [100, 104, 105, 101].

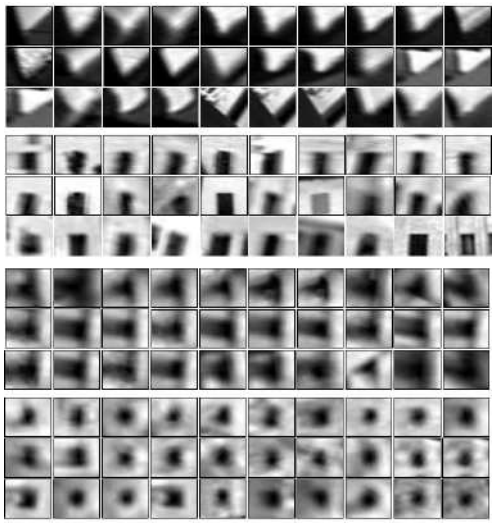
On the other hand, Implicit Shape Model (ISM) is a simpler pictorial structure representation that resembles a star-topology. In this model, the parts are modeled independently instead of a complex joint distribution. They are connected to each other indirectly or *implicitly* through the object center [102]. In the training phase, as the first step, a class specific local feature appearance vocabulary is created in order to model the appearance variation among the same object part instances. In the basic version of the implicit shape model method, the next step is accumulating votes for a histogram of object position hypothesis in order to utilize generalized Hough transform. The modes are searched in the hypothesis space and verification of the hypothesis is performed by backprojection of maximum for final evaluation [102]. In later versions of the model, as an intermediate step, object parts are defined as frequent co-occurrences of similar local patches at close locations in order to generate less ambiguous hypothesis. The model utilized in this approach is much more simpler to learn and test, and still it works well for many object categories with some useful extensions [106, 107, 108].

The common part in the constellation and ISM approaches mentioned above is the detailed utilization of spatial information.

There is another strand of research which decreases the priority of spatial information in order to stay robust against configuration variances. On the other hand, robustness against variance in imaging conditions is achieved via quantization of local appearance information in a way similar to the ISM. An illustration of appearance quantization is given in Figure 2.17(a). The widespread term for the list of quantized descriptors that are used to represent local appearance is the *visual codebook*. Each member of the codebook is called a *visual codeword*. These codewords, despite being illustrated in Figure 2.17(b) as being related to a prominent semantic part, need not necessarily have a human understandable meaning. These codewords are most often just a mere generalization of local appearance patterns that does not possess a meaning by themselves. In order to map these local appearance patterns to a higher level semantic entity like an object category, other generalizations are required.



(a)



(b)

Figure 2.17: Codeword Examples [109]. (a) Patches corresponding to the same codeword in a face detection problem, (b) Four groups of patches each corresponding to a seperate codeword in a generic image set.

The vanguard of local feature based category detection approaches, which prioritize appearance, is the *Bag-of-Words* paradigm. Bag-of-words (BoW) approach, in its most general form, represents images by a word frequency histogram [110], which has its roots in text document classification [111]. This frequency histogram is an accumulation of appearance statistics of visual words in the image (Figure 2.18). In the literature, this method has been utilized with variations in the appearance quantization, as well as the codeword assignment and histogram calculation methods [112]. For creating the appearance codebook, K-Means is the most com-

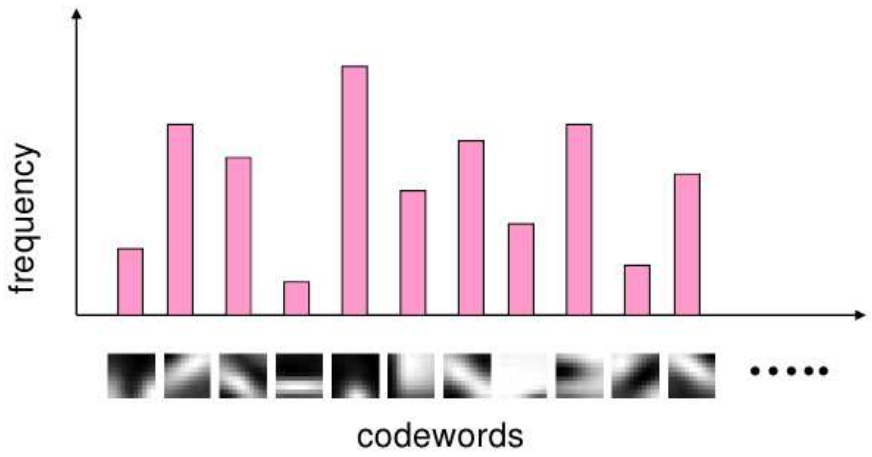


Figure 2.18: Visual Word (i.e. Codeword) Frequency Histogram [109].

mon approach with a lot of implementation variations [113]. Another widespread codebook creation method is Gaussian Mixture Model (GMM), which enables additionally enables the calculation of cluster membership likelihood for appearance descriptors [114]. After the creation of a visual codebook during the training stage, each local feature descriptor can be assigned a single codeword, as in the hard assignment case [115, 116], or can be a weighted member to multiple clusters, as in the soft assignment [117] The frequency histogram vector is obtained by one of these clustering and assignment combinations, and is used as the image representation for the following high level classification stage. In this stage, image representation vector is coupled with one of the histogram distance metrics, and used as input to a classifier for training and classification [118, 119, 120]. SVMs are widely accepted as the discriminative learning algorithm for histograms, due to the typical high dimensionality and sparsity of the descriptions [108, 120].

BoW methods are robust against position and orientation of object in image, since they discard the spatial information that is coupled with local features. They also have fixed description length irrespective of the number of local features, which is a favorable property when working with classifiers. As a summary, BoW approach is quite successful in classifying images according to the objects they contain. On the other hand, the lack of spatial information in BoW hinders it from localizing objects within the image, as well as from exploiting configuration of local features for reaching more specific categories of objects.

BoW model can be extended for better localization and dealing with the clutter problem, in a way similar to the sliding window approach to object recognition [58]. This extension in-

volves utilization of training data, in which the region of interest (ROI) of the object under consideration is known. BoW histograms of training data is extracted specifically from these ROIs, instead of the whole image. This approach, however, brings the requirement of extracting BoW descriptions of test images from the ROI of the objects to be detected. This chicken-egg problem is solved by a multi-scale sliding windows approach. BoW histograms are extracted from rectangular regions of varying sizes and positions in an image. Each of these ROI based description vectors are then fed to the classifier trained by the ROI-based training data [121]. Despite this ROI extension, the description still misses spatial information that is existent in the configuration of local features.

Spatial information can be incorporated into BoW approach at a moderate level by utilizing grid based approach for computing BoW histograms. In this approach, training images are utilized in the same way as in the previous ROI based approach. Additionally, ROI is divided into tiles using a spatial grid with an application specific resolution. Then, the BoW histogram is additionally indexed using the tile index (Figure 2.19) or in other words, concatenate histograms of tiles [122]. As the resolution of the grid increases, the length of the descriptor vector also increases. For instance, in case of a 2x2 grid, instead of a single length D histogram, four length D histograms with a total descriptor length of 4D is extracted. Grid-based descriptions of various resolutions can also be concatenated to obtain a representation similar to the spatial pyramid approach (Figure 2.20) utilized in the area of scene classification [123, 124]. It is also possible to assign larger weights to relatively complex, finer grid spatial BoW histograms during the histogram comparison step. Local features brings an additional invariance into the representation of objects by objectively and repeatably detecting *interesting regions*. For an approach resembling sliding windows, however, extracting multi-scale visual words on an overlapping dense grid also results in a plausible representation. The only difference here, is the utilization of BoW histograms obtained from dense local descriptors instead of sparse image fragments detected by local feature detectors. This extension of BoWs, which increases sampling detail in the expense of invariance, is utilized frequently in the literature [125, 115, 60, 122].

Object categorization methods were until recently only considering object region of interest for modeling, and therefore, detection. However, some common detection errors in standard methods are considered as symptoms of not using contextual information (Figure 2.21). Today, modern category detection methods are utilizing contextual information via scene de-

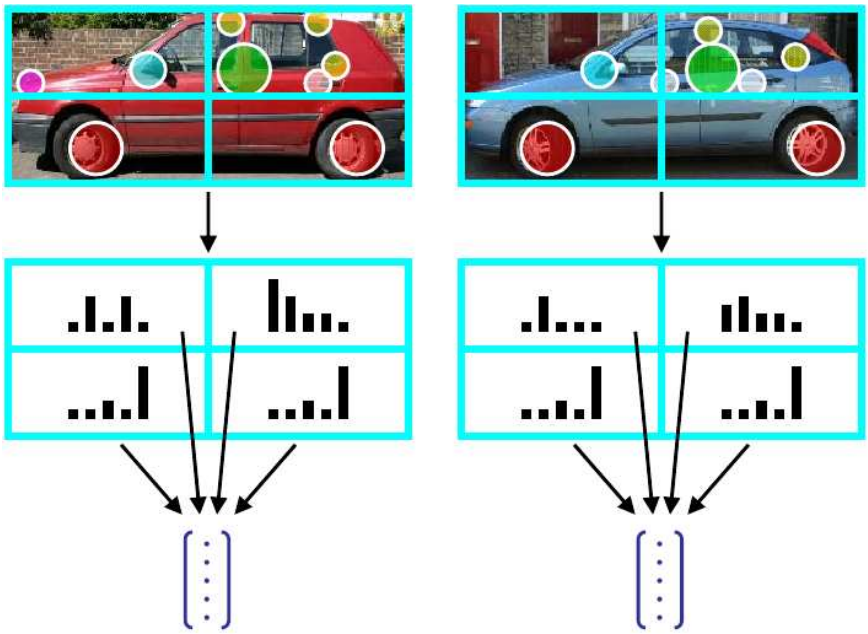


Figure 2.19: Grid-based Bag-of-Words Approach [109]: First compute bag-of-words histograms for each element of the spatial grid. Then, concatenate these independent histograms into a single feature vector to form the representation.

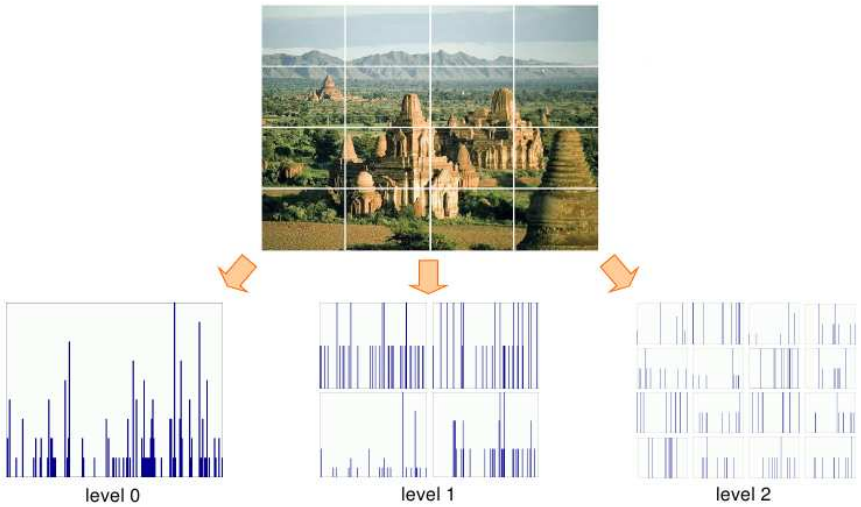


Figure 2.20: Spatial Pyramid Representation [124, 126]

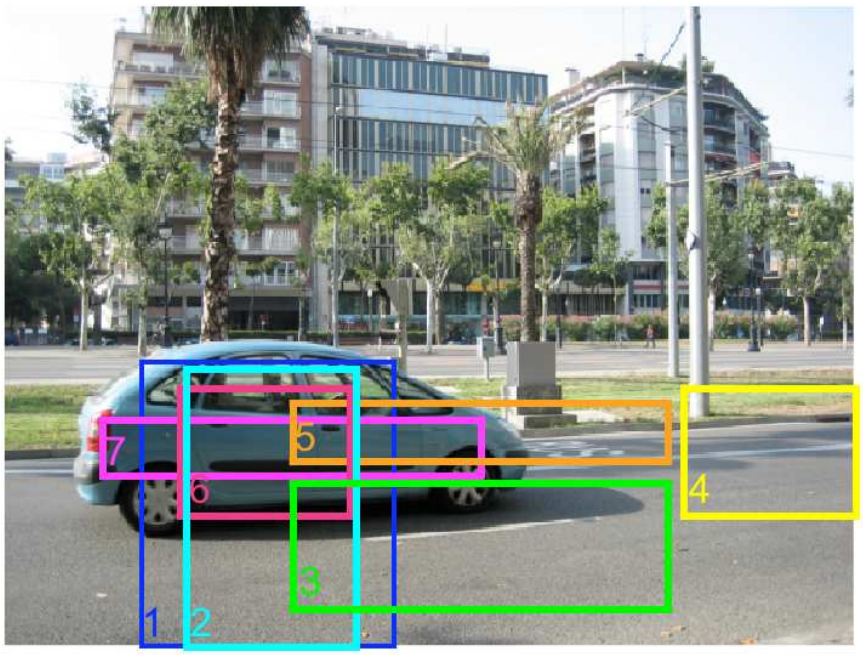


Figure 2.21: Example object detector results without context: 1. chair, 2. table, 3. road, 4. road, 5. table, 6. car, 7. keyboard [99].

scriptions, such as GIST [127], as a complementary information to local feature based models.

As an alternative to common supervised training approach, Malik introduced a different approach which involves utilization of low-level manually selected local features in a scheme called *Poselets* [128, 129]. The aim of this approach is to alleviate the flawless generalization problem by using strong supervision. Although, automatic local features are replaced with manually marked keypoint locations, this approach stands out as a novel research direction that may also be applied in the local features domain.

Between the two opposite poles of the local feature based object recognition application categories, namely wide-baseline matching and object category detection, stands the *instance level recognition* area. The term instance level is used in the sense that images are searched using a visual model database composed of instances (of object or scenes). Establishing correspondences is a necessity similar to the wide baseline matching case, however, here this goal is much more difficult to achieve due to the wider range of transformations that must be coped with. In Figure 2.22, large changes in scale, viewpoint and lighting along with partial occlusion are exemplified in a landmark recognition context.

At the current state of the art, most of the modern local feature-based methods that are devel-

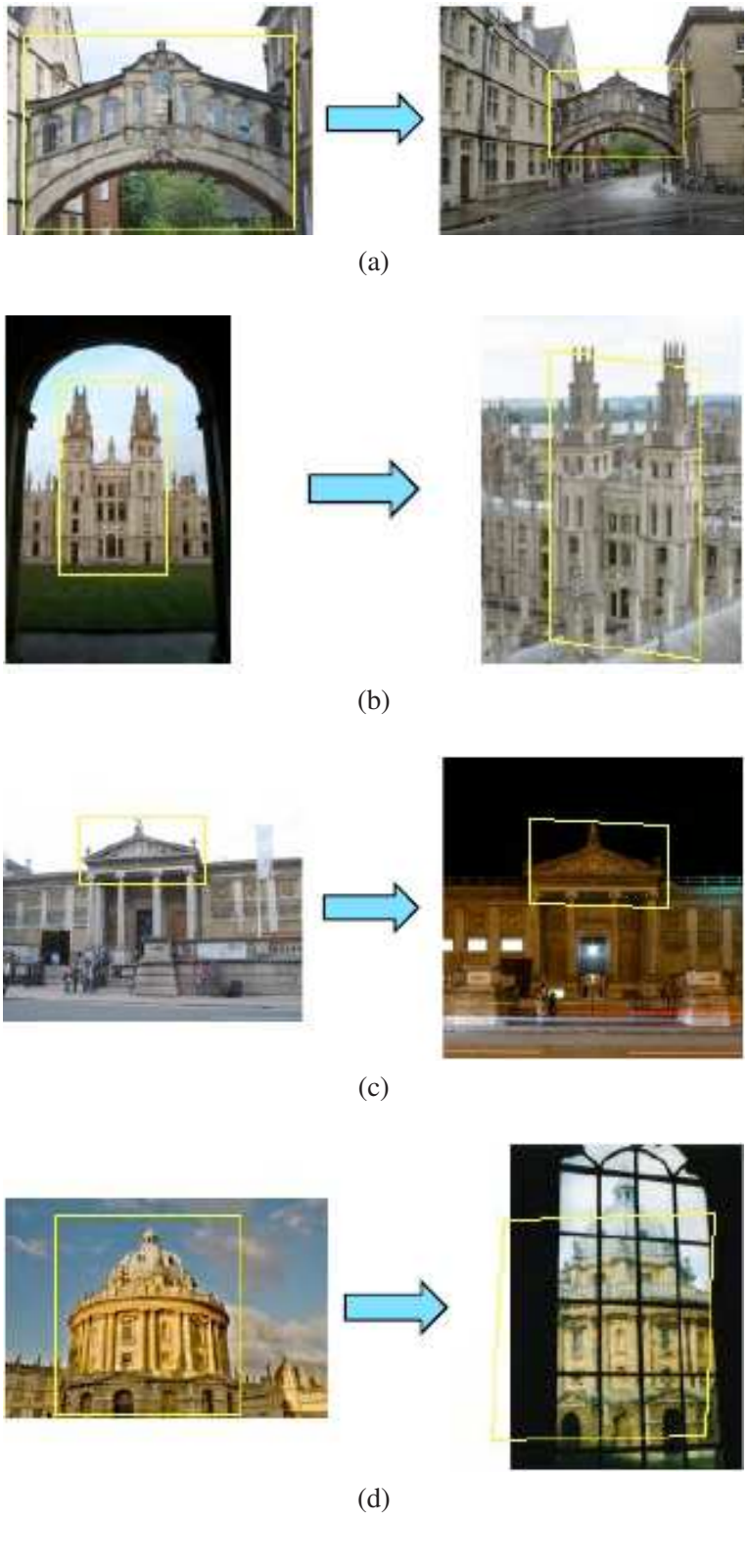


Figure 2.22: Common transformations in instance level recognition area (e.g. Landmark recognition) [126]. (a) Scale, (b) Viewpoint, (c) Lighting, (d) Occlusion.

Table 2.1: Time and memory requirements of nearest neighbor search for SIFT descriptors on a typical PC with 2.0 GHz CPU

Number of images	CPU Time	Memory Requirement
$N = 1$	0.4 second	128 kB
$N = 1000$	7 min	100 MB
$N = 10000$	1 hour 7 min	1 GB
$N = 10^7$	115 days	1 TB
$N = 10^{10}$	300 years	1 PB

oped for instance level recognition borrow a lot from wide baseline matching. In this matching approach, after the local feature detection and description steps, tentative correspondences based on the local descriptions of features are established individually (Figure 2.23), and then they are verified using geometric constraints. In order to establish tentative correspondences, one needs to solve a variant of the nearest neighbor search problem for all feature descriptor vectors x_j in the query image, among all the feature descriptors x_i in model images, which can be stated mathematically as

$$\forall j \; NN(j) = \arg\min_i \|x_i - x_j\| \tag{2.12}$$

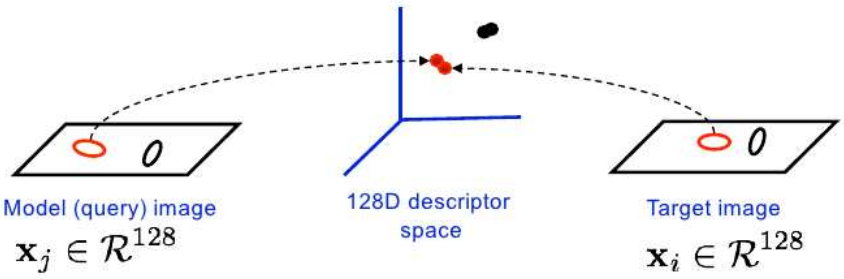


Figure 2.23: Tentative correspondence detection step using nearest neighbor search in the representative SIFT descriptor space [126].

Solving this problem might be easy for a limited number of target images; however, as the number of target images increase, time and memory requirements renders the task infeasible. This fact is illustrated in Table 2.1, where time requirements for nearest neighbor comparison of a query image with 1000 SIFT descriptors to another image on a single standart PC is used as a reference for projection.

As it can be seen from Table 2.1, nearest-neighbor matching constitutes a significant com-

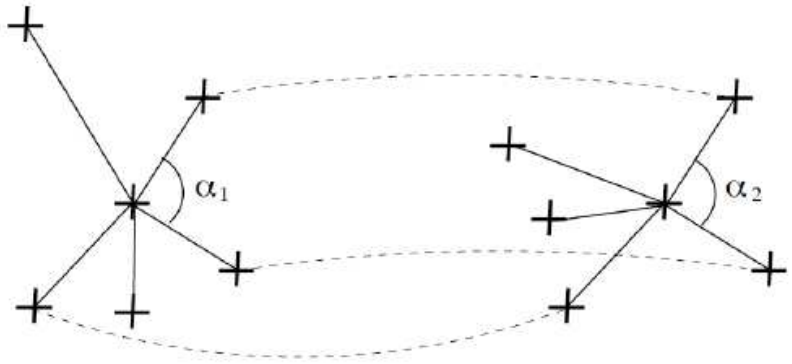


Figure 2.24: Angle-based geometric consistency constraints. (Left) A database entry and its p closest features, (Right) a match. Semi-local constraints: neighbors of the point have to match and angles have to correspond [41].

putational bottleneck. The complexity of a single linear nearest neighbor search for a local descriptor is $O(K^2ND)$ for a database with N images, each having K descriptors with a dimension of D. In the case of M model images this complexity becomes $O(MK^2ND)$. For typical values of D, the complexity problem can only be mitigated by approximate methods at the cost of an increase in failure rate. For approximate nearest neighbor search, hashing or tree based indexing approaches, such as kd-tree [130] can be utilized. Modern implementations of these approaches include optimizations for speed, as well as reduced failure rate. These implementations can be exemplified as Best-Bin First (BBF) [131], Approximate nearest neighbor kd-tree [132], randomized kd-tree [133] and Locality-Sensitive Hashing [134]. Comparison of modern versions of these methods can be found in [133].

For recognition methods adopting the philosophy of wide baseline matching, the next step in recognition is the verification of the tentative matches that are established in the previous step. This is achieved via a variety of methods using semi-local and/or global geometric constraints.

Semi-local constraints are based on matches that are located in a limited spatial neighborhood, while global geometry tries to fit a consistent relation between all of the matches. A prominent example to semi-local constraints is based on consistency of angles between the lines connecting a local feature to its nearest neighbors and their counterparts in the matched image [41, 135]. This constraint is visually illustrated in Figure 2.24. Surface contiguity filter [136] is another semi-local constraint, which is used to evaluate tentative local matches based on their consistency as a spatially continuous group (Figure 2.25).

Global constraints can also be used to evaluate the tentative matches which are the result of

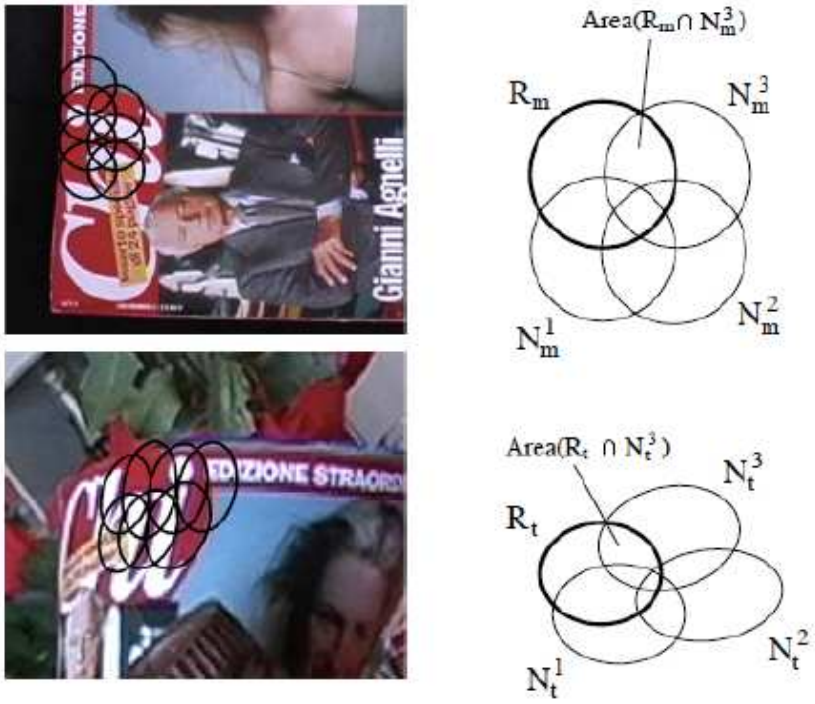


Figure 2.25: Surface contiguity filter. The pattern of intersection between neighboring correct region matches is preserved by transformations between the model and test images, since the surface is contiguous and smooth. The filter evaluates this property by testing the conservation of the area ratios [136].

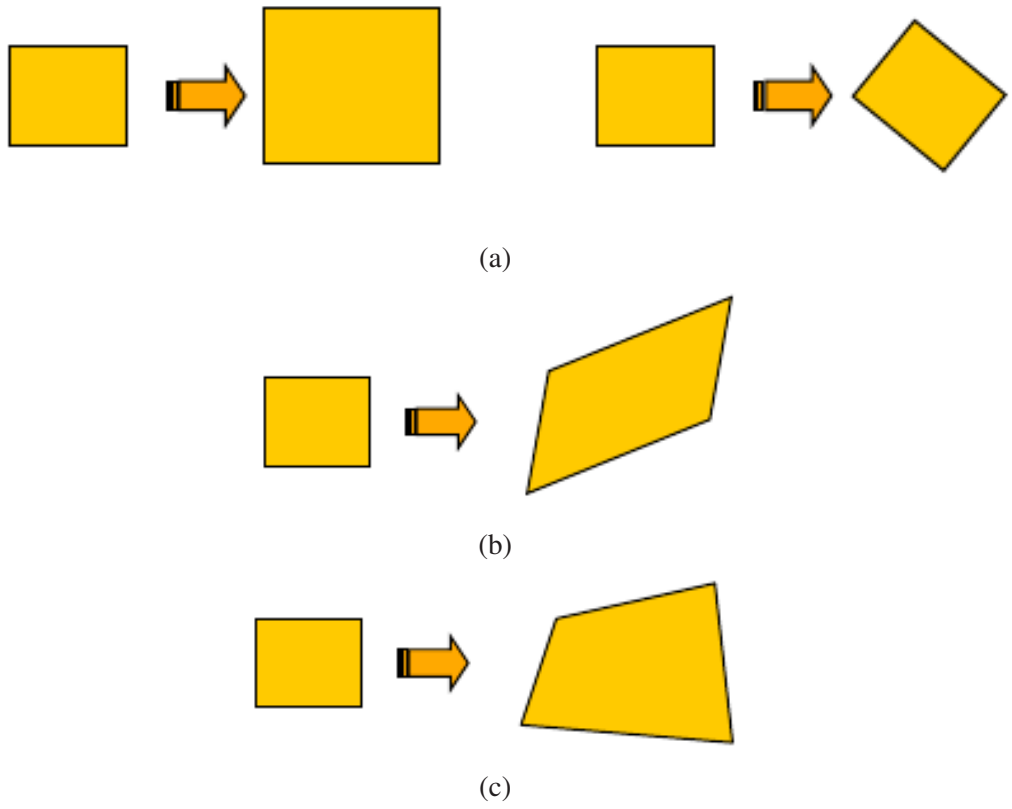


Figure 2.26: Transformation models for 2D Objects [126]. (a) Similarity Transformation (Translation, rotation. scale), (b) Affine Transformation, (c) Projective Transformation.

the local descriptor based independent similarity search step. Methods adopting this approach require the matches to be consistent with a single global geometric transformation. This transformation, however should also be estimated from the tentative matches. This example of the chicken-egg problem is generally solved by iterative methods. The type of the transform depends on the geometry of the queried entity as well as the assumed viewing conditions. For example, for a 2D entity, one of the three transformation models that are given in Figure 2.26 in increasing order of complexity, can be adopted. Note that each of these models can be represented by a 3x3 matrix called *homography* in homogeneous coordinates. In more general cases, where objects have 3D details, a more complex model can be used to define the one-way relationship between the real objects and their projections (Figure 2.27). Methods designed for 3D objects vary from that utilizing estimated 3D transformations for constructing 3D models [97] to ones using epipolar lines [137] induced by the transformation.

Estimating the transformation between two projections of an object with a set of tentative matches does not have a straightforward solution. This is due to the typically high ratio of

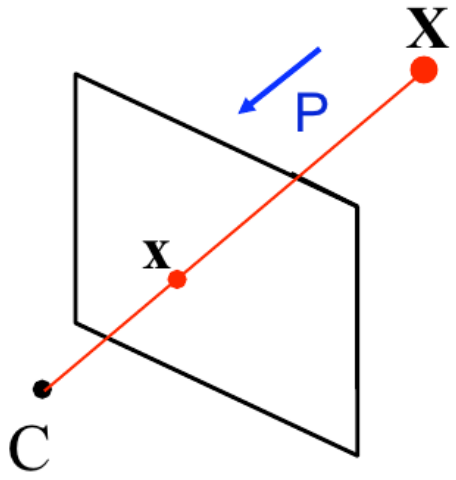


Figure 2.27: Generic 3D Projection Model [126]

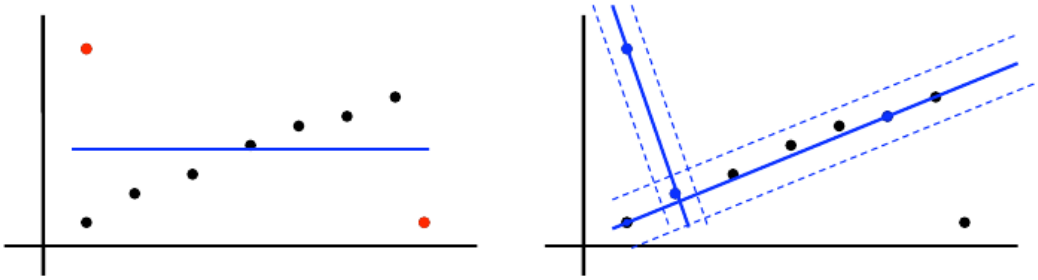


Figure 2.28: Line search in a point set using RANSAC [126]

erroneous matches (e.g. 90%) [75]. There are two main approaches to this problem, namely Random Sample Consensus (RANSAC) and Hough Transform. RANSAC is a robust estimation method [138] developed to deal with estimation problems in the presence of outliers. It depends on a characteristic loop that first selects a random seed group of matches. Next, a transformation is computed for this random group and then additional consistent matches, namely inliers, are searched. At each turn of the loop, a new transformation is estimated and if a transformation has enough inliers, it is refined via least squares estimation on its inliers. The transformation that has the highest number of inliers is selected as the final one. This process is illustrated in Figure 2.28. The most important parameter for RANSAC is the number of random samples, and analysis related to this issue can be found in [139]. Various extensions of this method for visual recognition applications also exist in the literature [140, 141].

The other widely accepted solution to global transform estimation problem is Generalized

Hough Transform [142]. Original version of Hough Transform is developed for detecting lines in a point cloud by searching for maxima in a line parameter voting space. Today, however, Hough transform is used at its generalized form where search space can be constructed from any parametrized entity. Global transformation estimation is an example to parametrized search spaces, where Hough transform is applicable. In the local feature based recognition area, Hough transforms most famous application is for estimating consistent matches among matches established using SIFT descriptor [75]. In this application, a planar affine transformation with six parameters is assumed between the matched images. Four parameters of this transformation, namely, scale, rotation, x and y translations are used as quantized dimensions of the voting space. Each tentative matching pair of scale and rotation invariant local features indicates an alignment hypothesis represented in these four parameters. These hypothesis are each allowed to cast votes to its eight closest bins in this coarsely quantized 4-dimensional voting space. Modes, or in other words, highly voted bins in the voting space, are then used for estimation of an affine transformation with six unknowns. These transformation estimates, which are further refined using inliers, constitute candidates for groups of final consistent matches. Hough transform and its extensions are utilized in many object recognition applications [75, 143, 102, 106, 144, 145], for extracting groupings from clutter in linear time.

Local features are equipped with different levels of invariance as explained in Section 2.1. Additionally, in most of the time, level of invariance is increased in the description phase via using appropriate descriptors (Section 2.2). The methods that adopt the classical approach, highly depend on the success of the initial matching of local descriptions. This matching operation, however, can be unacceptably expensive for target archives containing colossal numbers of description vectors (Table 2.1). Additionally, even this expensive operation can generate only tentative and mostly erroneous matches in typical cases [75]. This is due to the level of invariance incorporated into the descriptions, which already have reduced amounts of discriminative power due to the small sizes of their support regions. These methods, then try to filter the contaminated matching data using methods, like RANSAC or Hough Transform.

The methods that can utilize spatial information in the early matching phase of the local features have a significant advantage over brute force NN methods. They can avoid losing true matches during the early and unreliable NN matching phase, and they can utilize quantized descriptors for appearance consistency. One of the scarce methods, which incorporate spatial

information to the matching phase of local descriptions that is implemented by Quack et al sets a good example for novel methods of this kind [146, 147]. This method utilizes quantized appearance descriptors of local features by assigning them to a visual codebook. Next, each local feature is described in terms of the codewords of its neighbors. The novelty in this method is the addition of coarse relative location information to the codeword. Each local feature is described by a grid of predefined structure and size adaptive to its detected scale. The visual codewords of local features detected in this grid contribute to the binary codeword existence vector of the corresponding tile (Figure 2.29).

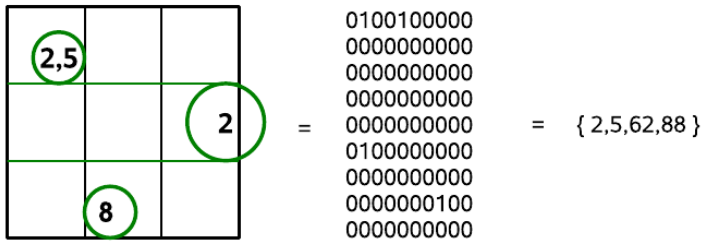
Binary descriptions of training data are analyzed using the *apriori* method [148], which is originally applied in the market basket analysis for finding associations among items. This way, frequent itemset configurations that are highly correlated with the concept under analysis are mined in the training set. These frequent itemset are then used in a weighted manner as cues of object presence in test images.

In this thesis, our approach to the object recognition problem is similar. Instead of trying to tweak the parameters and details of the current classical methods or using clever NN search and quantization methods, we try to incorporate the geometric invariance literature to the recent object recognition research that makes widespread use of local features. In this errand, we keep in mind that new methods need some time for becoming mature enough for practical large scale applications and hold on to the saying by Jitendra Malik [126]:

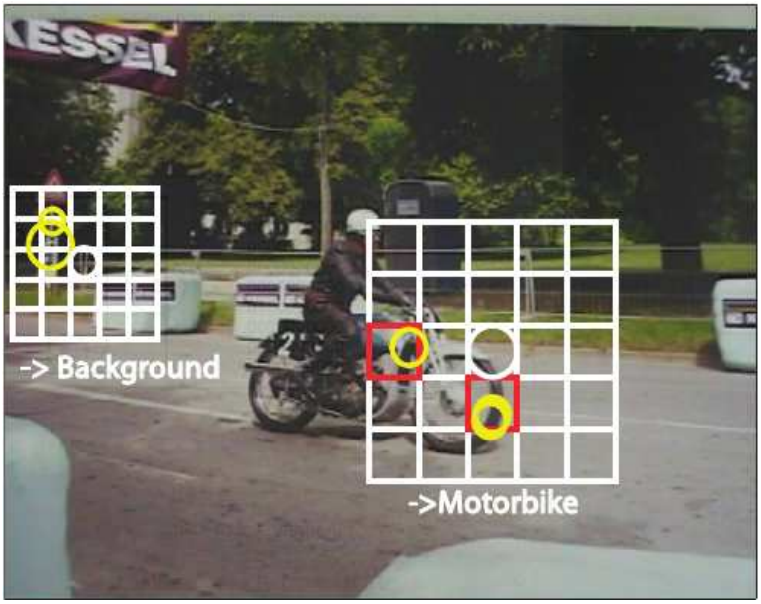
> We can't go to the moon building larger and larger ladders. And we will have to live with the fact that new methods work less good in the beginning.

We strongly believe that geometric invariants of local feature configurations that we will introduce in the next chapter can be used in mining strong cues of object presence in clutter. However, in order to exploit them, their integration with local appearance descriptions in a plausible way is required. This way, we may avoid solving the problem of correctly matching patches of objects and therefore finding correspondences based solely on local appearance information. In fact, this problem is much harder than the higher level problem of matching an object as a whole.

Our method utilizes some geometric invariants [149, 38, 150] to define groups of local features geometrically. Local appearance descriptions are also utilized in their basic forms along



(a)



(b)

Figure 2.29: Frequent Itemsets in object recognition [146, 147] (a) (Left) An example neighborhood with 9 tiles and 10 appearance clusters. Circles represent local features, and numbersindicate the appearance cluster(s) they are assigned to, (Center) Activation vector, (Right) Transaction, (b) Example of mined rules: (Left) A frequent configuration that is used to infer background, (Right) A configuration that is used to infer the object motorbike.

with simple quantization methods. The detailed optimization of quantization process is deliberately left out in order to assess the power of the proposed methods. In this manner, geometric descriptions are combined with appearance descriptions that are only suboptimal in terms of discriminative power between local patches. Detailed descriptions of these methods are examined in the following chapters and the results of simulations are also presented.

CHAPTER 3

INVARIANT GEOMETRIC RELATIONS AMONG LOCAL FEATURES

In this chapter, representation of 3D and 2D point coordinates in terms of homogeneous coordinates are first introduced. Next, hierarchies of 3D to 3D and 2D to 2D transformations related to this representations are analyzed. Camera models for approximating the 3D to 2D projection phenomenon and their relation to the transformations introduced in Section 3.1.2 are explained in Section 3.2. Geometric invariants of two types of representations, namely perspective and weak-perspective models, and related transformations, namely projective and affine space/transformations are introduced in the last section.

3.1 Elements of Geometric Coordinate Representations

In this chapter, the hierarchies of transformations (Section 3.1.2) and their relations to camera projection models (Section 3.2) are explained. In order to establish these relations, a consistent interpretation for measuring image coordinates and the position and orientation of geometric entities in an arbitrary coordinate system is required. This requirement is fulfilled in Section 3.1.1 by constructing a consistent set of coordinate representations for point coordinates in 3D and 2D.

In Section 3.3, invariants of transformations that take place in and between these coordinate systems are analyzed in detail. These invariants construct the basis for geometric descriptions that are proposed in Chapter 4.

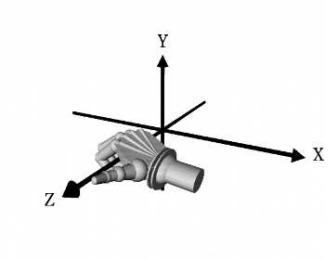
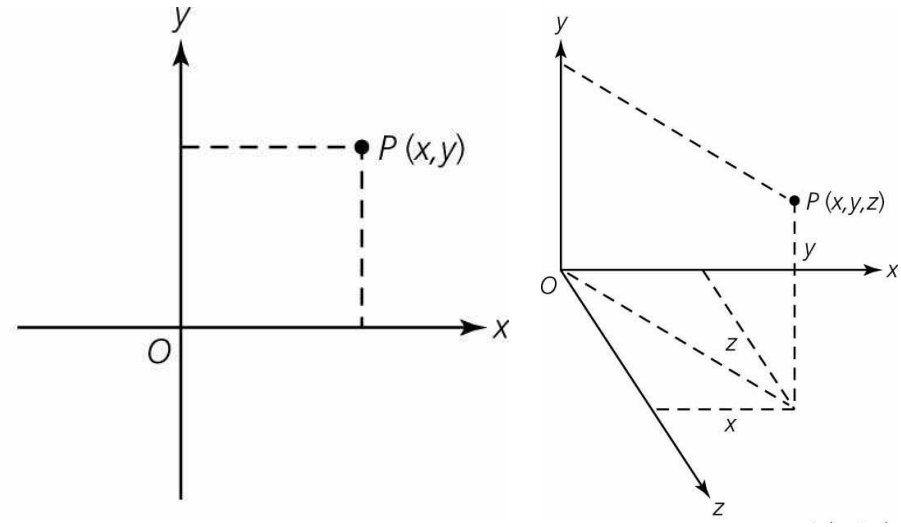


Figure 3.1: Right-handed Coordinate Frame Orientation [95]



(a) (b)

Figure 3.2: Coordinates of a point *P* in Cartesian coordinates [95]. (a) 2D Cartesian coordinates, (b) 3D Cartesian coordinates.

3.1.1 Homogeneous Coordinates

A three-dimensional *orthonormal coordinate frame (F)* can be defined by a point O in the physical three-dimensional Euclidean space E^3 and three unit vectors i, j, and k that are orthogonal to each other to be interpreted as *origin* and *basis vectors* respectively. Although, there are different conventions, the most common is the right-handed coordinate system which is illustrated in Figure 3.1.

Coordinates of a point P in 3D is then defined in this coordinate frame as x,y, and z, which correspond to the lengths of orthogonal projections of the vector $\vec{OP}$ onto the vectors i, j, and k, respectively. This fact is illustrated in Figure 3.2 for both 2D and 3D cases. The coordinates

of a point 3D can be represented as:

$$\vec{P} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \in \mathbb{R}^3 \tag{3.1}$$

Similarly, coordinates p of a point in 2D can be defined as illustrated in Figure 3.2:

$$\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \tag{3.2}$$

Homogeneous coordinates, on the other hand, represent points in a way that is especially useful in projective geometry which will be further discussed in the following sections. In homogeneous coordinates, the coordinate vector of a 3D point P in the same coordinate system (F) that is used above can be written as:

$$\vec{P} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.3}$$

by adding a fourth coordinate equal to 1 to the ordinary coordinate vector of P given in Equation (3.1). In homogeneous coordinates, a point P can only be defined up to scale since multiplying the coordinate vector in Equation (3.3) by a non-zero constant does not change the physical point that is referred by the coordinates (i.e. P). In order to go back from homogeneous coordinates to inhomogeneous coordinates, it is only required to divide the elements of the homogeneous coordinate vector by the fourth element and therefore normalize the fourth coordinate to one. In two-dimensional space, point coordinates can be represented in homogeneous coordinates similarly:

$$\vec{p} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3.4}$$

Although, the need for the homogeneous representation may not be clear at this point, it becomes vivid clear as soon as one needs to discriminate between affine and projective subspaces. This case is explained in the following sections.

3.1.2 Hierarchies of Transformations

Transformation models introduced in this section are characterized by their degree of freedom. These transformations are introduced by starting from the most constrained one and moving towards higher degrees of freedom. This variance in the degree of freedom, changes the properties that characterize the underlying transformations and subspaces. These characteristic properties remain unchanged under the related transformations and are closely related to the geometrical invariants that will be analytically derived in Section 3.3.

3.1.2.1 Transformation Models in Planar Case

The first subset of transformations of interest are transformations that take place in two-dimensions, namely, a plane. In two-dimensional space, a point P can be represented by a pair of coordinates (Equation 3.2), or alternatively, in homogeneous coordinates (Equation 3.4). The most general case of transformations in 2D, which is also called a *projectivity* can be defined as follows [95]:

Theorem 3.1.1 *A mapping $h : \mathbb{P}^2 \to \mathbb{P}^2$ is a projectivity, if and only if there exists a non-singular 3×3 matrix H such that for any point in $\mathbb{P}^2$ represented by a vector $\vec{x}$ it is true that $h(\vec{x}) = H\vec{x}$, where $\mathbb{P}^2$ represents 2D projective space.*

This relation interprets the homogeneous vector $\vec{x}$ as any point in $\mathbb{P}^2$ and $H\vec{x}$ as a linear mapping of homogeneous coordinates. The proof of this theorem can be found in [95] and will not be provided here. However, its results enable an alternative definition of the projective transformation as follows [95].

Definition 3.1.2 *A planar projective transformation is a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix:*

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{3.5}$$

or more briefly, $\vec{x'} = H\vec{x}$.

The most important thing about this definition is that the matrix H in Equation (3.5) may be changed by scaled by an arbitrary non-zero constant without changing the function of the projective transformation on the homogeneous coordinate vectors. As a result, one can say that H is a homogeneous matrix, since as in the representation of points in homogeneous coordinates (Equation 3.4), only the ratio of the matrix elements is important. There are eight independent ratios relating the elements of matrix H, and therefore, a projective transformation in 2D has eight degrees of freedom. A projective transformation in 2D, maps points from one plane to the other. However, it is useful to describe some specializations of projective transformation that model important phenomenon in the planar geometry.

There are four important categories of planar projective transformations, namely *Euclidean, similarity, affine* and the general *projective* transformations. Human language interpretations of the invariant properties valid for each of these categories are also important for a better understanding of these transformations. An *Euclidean* transformation can be represented as:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3.6}$$

where θ represents the angle of rotation and (t_x, t_y) represent translations in two axis directions. In a more easily interpretable way, Euclidean transformation can be rewritten as:

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} R & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.7}$$

where R is a 2×2 rotation matrix and $\vec{t}$ represents the translation on the plane. The most important property of transformations in the Euclidean category is their preservation of Euclidean distance. As a result, the Euclidean distance between two points transformed by a transformation of this category is invariant, and therefore, preserved in the transformed coordinate system.

A more general category of transformations in planar geometry is similarity transformation. It is an Euclidean transformation extended by scaling. This can be easily illustrated using the simplified notation in Equation (3.7):

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.8}$$

where the scalar s represents the scaling. A planar similarity transformation preserves the *shape*. In other words, similarity transformation preserves the ratios between the distances amongst points, although it changes the distance itself.

The categories of transformations that we have analyzed so far represent physical transformations that can be applied to entities without distorting their shapes like rotation, translation and scaling. The following two transformations, however, represent distortions that arise from more complex and hard-to-interpret operations.

Affine transformations, or in short *affinities*, can be represented by a more general block matrix form:

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} A & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.9}$$

where the only requirement on A is being a non-singular 2×2 matrix. Distortions arising from an affine transformation can be better understood by a more intuitive representation:

$$A = R(\theta)R(-\phi)DR(\phi) \tag{3.10}$$

where $R(\theta)$ and $R(\phi)$ represent rotations and D represent a diagonal matrix which performs scaling with respect to an intermediate orthogonal coordinate frame. The effect of an affine transformation is illustrated in Figure 3.3.

The geometrical properties that remain invariant under affine transformation are parallelism, ratio of lengths of parallel line segments and ratio of areas. The latter invariant property is closely related to the invariant algebraic descriptions in Section 3.3.

In its most general form, or with highest degrees of freedom, projective transformations can be represented as:

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.11}$$

The most important property of projective transformations is that they do not preserve parallelism, i.e. two parallel lines represented in one planar projective coordinate frame may be intersecting in the transformed frame. In fact, in the projective space, two lines that are parallel are assumed to intersect at points called *ideal points*. These points represent points at infinity and are used to illustrate the abstract fact that in projective space every line pair intersects at least one point. With this augmented definition, we can say that general projective transform preserves incidence relations like the order of intersection between lines.

The general unconstrained projective transformations, as previously stated, has eight degrees of freedom. The other categories of transformations namely, affine, similarity and Euclidean have six, four and three degrees of freedom, respectively. The number of degrees of freedom is directly related to the number of numerical invariants of a transformation. The hierarchy of planar transformations are illustrated visually in Figure 3.3.

3.1.2.2 Transformation Models in 3D Case

Points in 3D space are represented in homogeneous coordinates using a four dimensional vector as previously stated in Section 3.1.1. Similar to the planar case (Equation 3.5), a projective transformation in $\mathbb{P}^3$ can be represented as a linear transformation on homogeneous four dimensional vectors, which is defined by a 4×4 matrix:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \tag{3.12}$$

This definition can be simplified as, $\vec{x'} = H\vec{x}$ where H is a 4×4 non-singular matrix. This matrix acting on 4-vectors is homogeneous, as the vectors it acts upon, and therefore, has 15 degrees of freedom corresponding to the number of independent ratios amongst its elements.

There are four important categories of projective transformation of 3-space or 3D points.

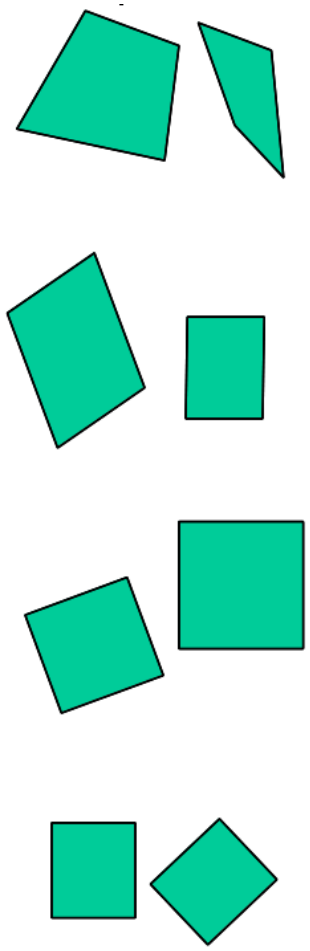


Figure 3.3: Hierarchy of Planar (2D) Transformations. Bottom-to-top: Euclidean, Similarity, Affine, Projective.

These are analogous to their planar counterparts, and therefore, can be represented by similar characteristic matrices. An Euclidean transformation in 3D is represented by the following equation:

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} R & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.13}$$

where R represents a 3×3 rotation matrix, and $\vec{t}$ represents a 3×1 translation vector.

Similarity transform that is a scale augmented version of Euclidean is represented by:

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.14}$$

where s represents a constant scalar for uniform scaling.

Non-uniform scaling characterizes the affine transform, which can be represented as:

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} A & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.15}$$

where A represents any 3×3 non-singular matrix without any other constraints.

Lastly, the superset of all the preceding transformations, general projective transformation is represented by:

$$\begin{pmatrix} \vec{x'} \end{pmatrix} = \begin{bmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} \tag{3.16}$$

where $\vec{v}$ is a 3×1 vector representing perspective distortion effects that characterizes this category.

The effects of the four categories of transformations are illustrated visually in Figure 3.4

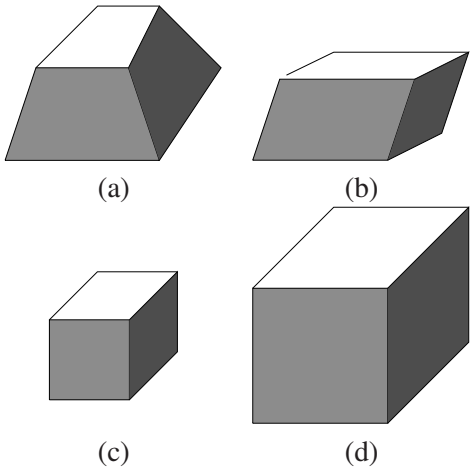


Figure 3.4: Hierarchy of 3D Transformations. (a) Projective, (b) Affine, (c) Similarity, (d) Euclidean [95].

3.2 Geometric Camera Models and Constraints on Approximations

Camera models are used to represent the mapping from 3D space to its 2D image. Although 3D world is assumed to stay the same during projection, each set of camera parameters representing various aspects of projection, such as position, principal axis direction and angle between the imaging plane axis results in a different 2D image. Since our major goal is to use invariants of a subset of transformations, we assume the camera model that suits our assumptions.

3.2.1 Perspective Camera Model

The most simple camera model is the pinhole camera, which is illustrated visually in Figure 3.5.

According to the pinhole camera model a general perspective camera model can be represented mathematically as:

$$\vec{x} = P\vec{X} \tag{3.17}$$

where P represents the 3×4 camera projection matrix from a world point represented by a homogeneous 4-vector $\vec{X} = (X, Y, Z, 1)^T$, to an image point represented by a homogeneous 3-vector $\vec{x}$.

In a simplified pinhole camera model (Figure 3.5), where the camera center is assumed to be

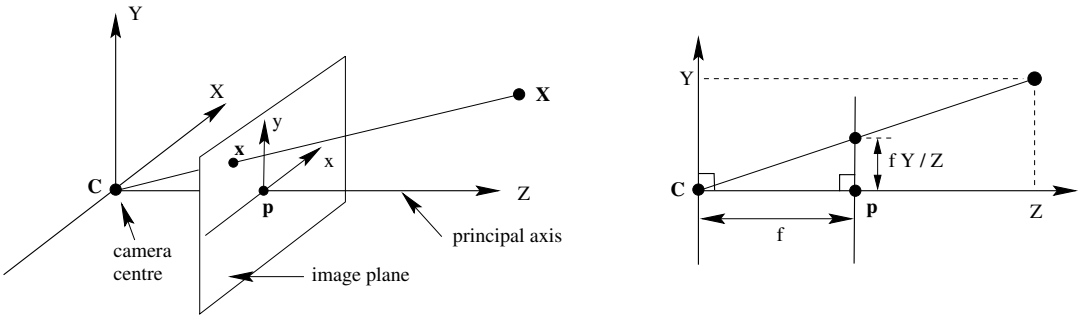


Figure 3.5: The Pinhole Camera Model. C is the camera center and p is the principal point. The camera center and the coordinate system origin coincide. Image plane is placed in front of the camera [95].

located at a distance f from the origin of the image plane, whose axis correspond with the camera coordinate frame and the principal axis of the camera aligned with the z-axis can be defined as:

$$P = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.18}$$

In order to stress camera projection model assumptions the projection matrix P can be expressed in a more intuitive form that is:

$$P = K \left[I \mid \vec{0} \right] \tag{3.19}$$

where K is called the camera calibration matrix representing the internal parameters of the camera, and I is a 3×3 identity matrix. Camera calibration matrix here is defined as:

$$K = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \tag{3.20}$$

In a typical finite projective camera, this matrix is defined as follows [95]:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{3.21}$$

where α_x and α_y represent the Charge Coupled Device (CCD) pixel sizes, s represent the skew parameter modeling non-orthogonality of the camera axis, and (x_0, y_0) represents the offset between the principal point and the origin of the imaging plane. In addition, this matrix should be non-singular in order for the camera to be finite.

External parameters of a camera are represented by a matrix that substitutes the augmented matrix $\left[I \mid \vec{0}\right]$ in Equation (3.19). This matrix represents the coordinate system change between the world coordinate frame and the camera coordinate frame; in other words, the viewpoint of the camera. Including this matrix, the finite projection matrix is defined as:

$$P = K\left[R \mid \vec{t}\right] \tag{3.22}$$

where R represents the world coordinate frame orientation with respect to the camera coordinate frame and $\vec{t}$ represents the world coordinate frame origin in the camera coordinate frame as $\vec{t} = -R\vec{C}$.

In a finite camera projection, perspective effects that are modeled by a projective transformation in 3D (Section 3.1.2.2) and 2D (Section 3.1.2.1) are typically observed. The most common effect is the mapping of parallel world lines to converging lines in the image (Figure 3.6).

$$P = H_{2P}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} H_{3P} \tag{3.23}$$

where H_{2P} and H_{3P} are 3×3 and 4×4 projective transformation matrices representing transformations in 2D and 3D, respectively.

The most prominent effect of a finite projection camera is the perspective effect. Imaging process under the conditions, where the distance of world points from the camera creates dominant effects on the resultant images are therefore frequently called as *perspective projection*.

(a) (b)

Figure 3.6: Effects of perspective distortion. (a) Orthogonal view of floor tiles, (b) Floor tiles image obtained with perspective projection [95].

3.2.2 Weak-Perspective Camera Model

The finite projective camera model, which models the typical projection process without any assumptions is defined in Section 3.2.1. Although this model is useful with a wider variety of imaging conditions, there are some significant simplifications that may be applied in special cases. One of the most useful of these simplifications is the *weak-perspective* camera model.

As its name indicates, weak-perspective camera model approximates the subset of projection conditions under which perspective effects diminish. An example of imaging conditions, where the weak-perpective model provides a good approximation, is illustrated in Figure 3.7.

Weak-perspective model approximation errors diminish, and the model gives results that are close to the finite projective model under these 3 factors:

- Increasing focal length f of the camera
- Negligible depth relief (δ) in the scene when compared to the average distance of the scene from the camera (d_0)
- Smaller distance of the imaged point from the image center

The error experienced when using a weak-perspective camera can be best illustrated visually as in Figure 3.8.

The weak-perspective projection model is represented by the following projection matrix which is to be substituted in Equation (3.17):

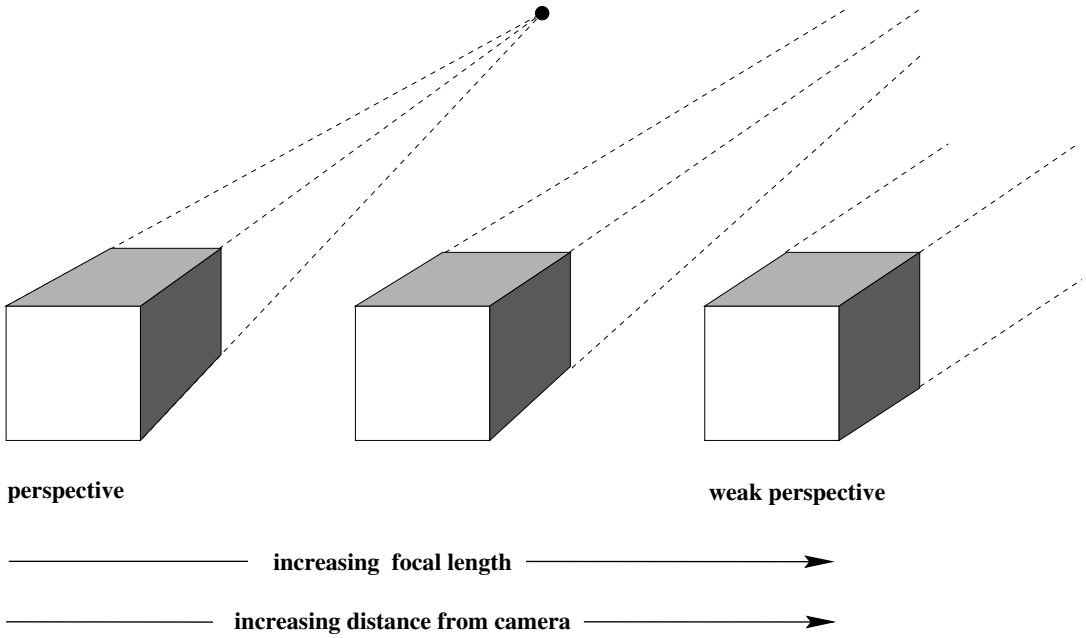


Figure 3.7: Imaging conditions where weak-perspective model is valid. From left to right, camera focal length and average scene distance from the camera increases. Note the re-emergence of parallelism [95].

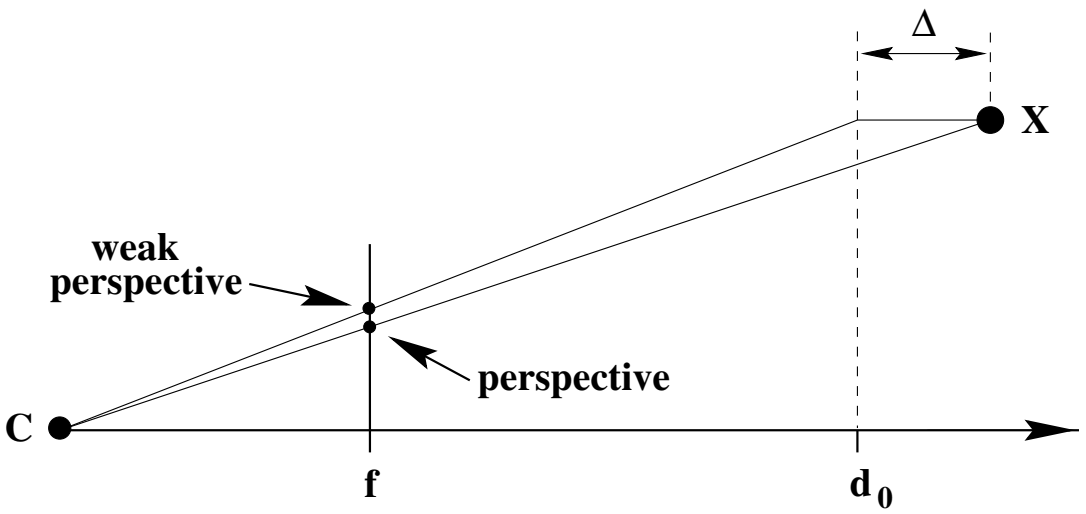


Figure 3.8: Error experienced when using a weak-perspective camera. d_0 is the average scene depth, C is the camera center, f is the focal length, X is the projected point, and Δ represents the depth relief of point X from the average scene depth [95].

$$P = \begin{bmatrix} \alpha_x & & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} \vec{r1}^T & t_1 \\ \vec{r2}^T & t_2 \\ \vec{0}^T & 1 \end{bmatrix} \tag{3.24}$$

Weak-perspective projection is a special case of the abstract *affine projection model* and can be represented in terms of its affine components. This can be performed by rewriting the weak-perspective projection matrix in terms of a 2D to 2D affine transformation, a projection from 3D to 2D and a 3D to 3D affine transformation:

$$P = H_{2A} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} H_{3A} \tag{3.25}$$

where H_{2A} and H_{3A} are 3×3 and 4×4 affine transformation matrices representing transformation in 2D and 3D, respectively.

The main difference between the weak-perspective and perspective projection models is the relative insensitivity of the former model to the intra-scene depth differences as illustrated in Figure 3.8.

3.3 Geometric Invariants of Local Feature Groups

Geometric invariants are properties that are specific to the underlying structure of an object or scene in the real world, and therefore, can be measured invariably under different viewing conditions. The invariants are characterized by two properties [149]. The first property is the dimensionality of the underlying object structure. 2D (or planar) objects can be characterized by 2D invariants, while 3D objects can be represented by 3D invariants. The second property is the level of invariance. Invariants are designed according to the requirements of the problem they are designed to solve. For instance, objects that undergo projective transformations need to be defined in terms of projective invariants, while for the objects that undergo affine transformations, affine invariants suffice.

Although invariants of more general types of transformations are also invariant under subsets of these transformations (i.e. Projective invariants are also invariant against affine transforma-

tions), considering the right level of invariance is still important. This is due to the increasing complexity and support size of the invariant calculations. Invariants that are robust against more generic types of transformations are defined in higher degree terms, and therefore, more fragile in the presence of errors in these terms. In addition, higher level of invariance is achieved only by increasing the support size, in other words, the amount of data required to compute the invariant.

This chapter first introduces invariants of 2D structures under affine and projective transformations in Section 3.3.1. Next, invariants of 3D structures under affine and projective transformations are introduced in Section 3.3.2. These invariants constitute the basis for local feature based geometric descriptions that are developed in Chapter 4 and Chapter 5.

3.3.1 Invariants of 2D to 2D Transformations

In this section, invariants representing geometric characteristics of 2D or planar objects are explained. Two types of invariance, namely affine and projective are analyzed, due to their frequent occurrence in common problems.

3.3.1.1 Affine 2D Invariants - Barycentric Coordinates

In order to explain affine invariants, it is a prerequisite to explain the vector space in which they exist. In this context, we are interested in objects that exist on a plane. A plane is a subspace, which can be defined in terms of three noncollinear points. For instance, consider three points A, B and C in $\mathbb{R}^3$ affine space. These points define the plane $S(A, B, C)$, in which any point can be uniquely defined in terms of the coordinates of points A, B and C. The coordinates of these points can be defined in terms of two vectors $\vec{u_1}$ and $\vec{u_2}$, which correspond to $\vec{AB}$ and $\vec{AC}$, respectively. These affine coordinates are actually a generalization of Euclidean coordinates where basis vectors $\vec{u_1}$ and $\vec{u_2}$ must be orthonormal [151]. Affine coordinates are also called *barycentric* coordinates in the literature [151].

Barycentric coordinates can also be interpreted in a more intuitive way using invariance of area ratios under affine transformation (Section 3.1.2.1). In this interpretation, Barycentric coordinates are represented in terms of a triangular representation (Figure 3.9)

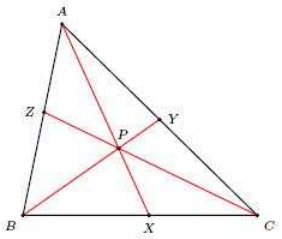


Figure 3.9: Barycentric coordinates visualization. P : the point whose Barycentric coordinates are computed; A, B, C : the basis points

$$\alpha = \frac{\triangle PBC}{\triangle ABC} \quad \beta = \frac{\triangle PCA}{\triangle ABC} \quad \gamma = \frac{\triangle PAB}{\triangle ABC} \tag{3.26}$$

where δ denotes the area of the triangle. The number of degrees of freedom in 2D affine transformation is six (Section 3.1.2.1), and therefore the number of invariants that can be obtained by the four points (P, A, B, C) can be computed as:

$$Number\ of\ Invariants = 2 \times 4 - 6 = 2 \tag{3.27}$$

Indeed, barycentric coordinates of a 3-point basis in 2D actually has 2 independent components due to the constraint that $\alpha + \beta + \gamma = 1$, which is obvious from the intuitive triangle area ratio example in Figure 3.9.

Distance between two barycentric coordinates, $\vec{x}$ and $\vec{y}$ of dimension 3 can be computed by the following metric, whose result is normalized to $[0.0, 2.0]$:

$$d = \frac{\left|\vec{x} - \vec{y}\right|}{\max(|\vec{x}|, |\vec{y}|)} \tag{3.28}$$

where $|\cdot|$ represents Euclidean (L_2) norm.

Change in barycentric coordinates for a typical four point configuration is given in Fig. 3.10. Simulation coordinates of the 2D points, A, B, C and P that are previously illustrated in Fig. 3.9 are given in Table 3.1. In order to analyze the change of the barycentric coordinate distance with respect to the error in point coordinates, x and y coordinates of point P are simultaneously perturbed in the range $[-1.0, 1.0]$.

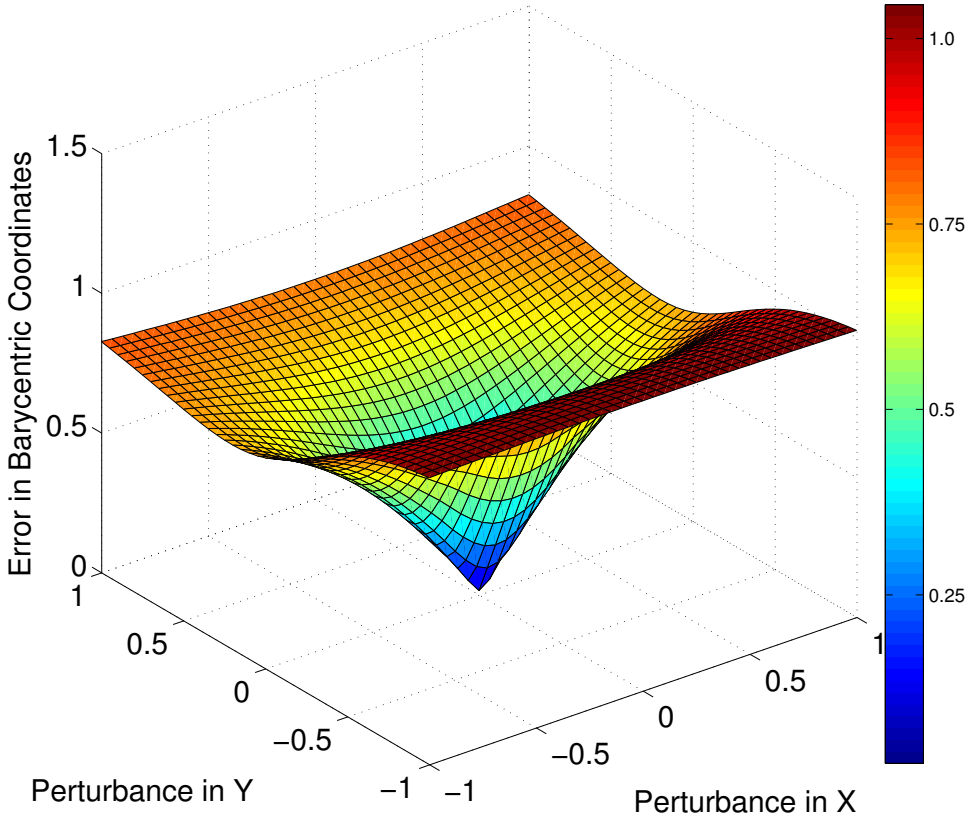


Figure 3.10: Effect of 2D coordinate error on barycentric coordinates for a typical four point configuration A, B, C and P, whose coordinates are given in Table 3.1. x and y coordinates of point P is perturbed in the range $[-1.0, +1.0]$ and barycentric distance of the new configuration to the original one is computed.

Table 3.1: Coordinates of four 2D points used for barycentric coordinate distance simulation

	2D Coordinates of Simulation Points				
	A	B	C	P	ΔP
x	0	-1	1	0	[-1.0, +1.0]
y	1	0	0	0.5	[-1.0, +1.0]

3.3.1.2 Projective 2D Invariants

Projective invariants are an extended version of affine invariants, which is robust against a larger category of transformations, that have 8 degrees of freedom. Therefore, their definition requires more than four points, which is the requirement for 2D affine invariants. Indeed 2D projective invariants require a 4 point basis to define invariant coordinates of the fifth point in terms of them to obtain $2 \times 5 - 8 = 2$ invariants.

Determinants are relative invariants of both projective and affine transformations. Using this fact, we define invariants in terms of the cross ratios of these determinants. Affine invariants that are derived in the previous section (Section 3.3.1.1) can also be defined in a similar manner, but using ratios of determinants instead of their cross ratios.

Any three of five points in 2D space that are represented in homogeneous coordinates can form a determinant. In addition, determinant of a matrix multiplication also have the following algebraic property (see Appendix A):

$$|TM| = |T|\,|M| \tag{3.29}$$

Using the above algebraic property, we can prove that cross ratios of determinants formed by five points a, b, c, d, e and their transformed versions A, B, C, D, E, which are obtained by applying a 3×3 projective transformation T, are invariant. In Equation (3.30), determinants $|ACD|, |ADE|, |ABD|$ and $|ACE|$ are computed from matrices formed by three transformed points, while $|acd|, |ade|, |abd|$ and $|ace|$ are computed from the matrices formed by original 2D points.

$$\frac{|ACD|\,|ADE|}{|ABD|\,|ACE|} = \frac{|acd|\,|ade|}{|abd|\,|ace|} \tag{3.30}$$

There are other arrangements of determinants, from which a total of two are independent for

a set of five points. These invariants exist only for five points, no two of which are the same and no three of which are collinear [151].

3.3.2 Invariants of 3D to 3D Transformations

Any point in 3D coordinates represented by a 4-vector can be defined as a linear combination of four other points as [149]:

$$\vec{X_5} = a\vec{X_1} + b\vec{X_2} + c\vec{X_3} + d\vec{X_4} \tag{3.31}$$

In this representation, the coefficients (a, b, c, d) are unique for affine space, while they can be multiplied by an arbitrary factor in the projective space. Invariants under two different categories of transformations in 3D are therefore obtained using different formulations.

3.3.2.1 Affine 3D Invariants

Any point in 3D coordinates represented by a 4-vector can be defined uniquely as a linear combination of four other points as in Equation (3.31). Similarly, determinants that are relative invariants of affine and projective transformations are utilized to obtain affine invariants.

It is proved in [149] that three of these affine invariants are independent, and they can be computed using determinants, which are also relative invariants of projective transformations. In order to do so, denote determinant of the 4×4 matrix formed by an ordered quadruple of 3D point coordinates, as M_i. For instance, determinant of the matrix formed by the first four points, $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4})$, can be defined as:

$$M_5 = \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| \tag{3.32}$$

Using this determinant definition, 3D affine invariants of five points can be obtained as:

$$I_1 = \frac{M_1}{M_5}, \quad I_2 = \frac{M_2}{M_5}, \quad I_3 = \frac{M_3}{M_5} \tag{3.33}$$

Proof of these invariant definitions can be found in Appendix A.

3.3.2.2 Projective 3D Invariants

Points in projective 3-space, which are represented by homogeneous coordinates can not be uniquely defined by Equation (3.31). Instead, the coordinates (a, b, c, d) can be multiplied by an arbitrary non-zero factor, still satisfying the equation. Therefore, this factor need to be eliminated in order to obtain projective 3D invariants.

In order to obtain invariants, at least six points are needed according to the number of invariants calculation ($3\times6-15 = 3$). Invariants of six points are derived following the calculations in Equation (3.33) for points $\vec{X_5}$ and $\vec{X_6}$ and using cross ratios to eliminate the constant factors. Points $\vec{X_5}$ and $\vec{X_6}$ are defined in the same way as affine case:

$$\lambda_5\vec{X_5} = a\lambda_1\vec{X_1} + b\lambda_2\vec{X_2} + c\lambda_3\vec{X_3} + d\lambda_4\vec{X_4} \tag{3.34}$$

$$\lambda_6\vec{X_6} = a'\lambda_1\vec{X_1} + b'\lambda_2\vec{X_2} + c'\lambda_3\vec{X_3} + d'\lambda_4\vec{X_4} \tag{3.35}$$

Two sets of unknowns $a\frac{\lambda_1}{\lambda_5}, \cdots d\frac{\lambda_4}{\lambda_5}$ and $a'\frac{\lambda_1}{\lambda_6}, \cdots d'\frac{\lambda_4}{\lambda_6}$ exist in two sets of four equations. We can eliminate the λ_i in the equations above by using cross ratios of determinants M_i, and obtain three projective invariants using coordinates of points $\vec{X_5}$ and $\vec{X_6}$ as:

$$I_1 = \frac{ab'}{a'b} = \frac{M_1M_2'}{M_1'M_2}, \quad I_2 = \frac{ac'}{a'c} = \frac{M_1M_3'}{M_1'M_3}, \quad I_3 = \frac{ad'}{a'd} = \frac{M_1M_4'}{M_1'M_4} \tag{3.36}$$

M_i' terms (based on the definition in Equation (3.32) in the above equation represent determinants of quadruples from the point set $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_6})$, instead of $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5})$ [149].

3.3.3 Invariants of 3D to 2D Transformations

This type of transformation is also known as projection, and is already analyzed in Section 3.2. In projection from 3D to 2D, the depth information is lost. This loss of information inhibits the development of invariants of a projection, even though 3D to 3D invariants and 2D to 2D invariants do not have this problem. The fact that no geometric invariants exist for a projection from 3D to 2D has been proved by Burns et. al in [36].

CHAPTER 4

JOINT UTILIZATION OF APPEARANCE AND GEOMETRY FOR PLANAR OBJECT RECOGNITION

In this chapter, a novel approach that is based on utilizing the competence of interest point detectors to capture primitives and the capability of geometric invariants to discriminate between spatial configurations of these primitives is presented. In the proposed approach, geometric constraints are enforced by means of 2D affine geometric invariants. These invariants are utilized as a geometric description of a group of interest points. Using this description, local appearance descriptor based potential matches are filtered and evaluated according to their geometric description as a group. All three novel methods that are introduced in this chapter share this common philosophy.

Main components of these methods are comparatively summarized in Table 4.1. A prominent and widespread approach by Lowe [75] that is also included in this table, is implemented and utilized as a reference for experimental evaluation of the proposed methods of this chapter. This baseline method is explained in detail in Algorithm 4.1.

The first part of this chapter, Section 4.1, describes the application of this hybrid local feature configuration representation scheme to one-to-one image matching. This method combines local feature appearances described by SIFT descriptor with the geometric invariants of 2D affine transformation, also known as barycentric coordinates (Section 3.3.1.1). The main aim of the method is to achieve correspondences between different views of semantically relevant partially planar objects and assess the performance of the overall approach in a constrained environment. The details of this one-to-one matching method is given in Section 4.1.2. Simulations of this method are performed on a constrained dataset containing partially planar objects. Experimental results presented in Section 4.1.3 show the robustness of the joint rep-

Table 4.1: Main Components of Planar Object Recognition Methods

	Appearance Matching	**Geometric Verification**	**Applicability**
Hough-based (Alg. 4.1) [75]	DoG & SIFT (Section 2.1.2 & 2.2) Best Match Only	Hough Trans. Clust. 2D Aff. Trans. Est. (Algorithm 4.1)	Planar Objects Single instance only Limited appearance distortion
Proposed Method 1 (Sec. 4.1)	SURF & SIFT (Section 2.1.2 & 2.2) Potential Match List	Barycentric Coords. (Section 3.3.1.1)	Planar Objects Multiple instances Moderate appearance distortion Limited scale changes
Proposed Method 2 (Sec. 4.2)	DoG & SIFT (Section 2.1.2 & 2.2) Quant. Appearance. (Visual Codebook)	Barycentric Coords. (Section 3.3.1.1)	Planar Objects Multiple instances Significant appearance distortion Limited scale changes
Proposed Method 3 (Sec. 4.3)	DoG & SIFT (Section 2.1.2 & 2.2) Quant. Appearance (Visual Codebook)	Barycentric Coords. (Section 3.3.1.1) Significance-based Grouping . via BG Density Est. (Section 4.3.2.2)	Planar Objects Multiple instances Significant appearance distortion Significant scale changes

resentation and the matching method against appearance variations, which severely degrade the performance of local appearance descriptions extracted from small patches. This one-to-one matching approach has been published in [47].

The second part of this chapter, Section 4.2, presents another method, which extends the preliminary work described in Section 4.1 in various ways. First, we replace the initial full descriptor vector-based appearance description stage with vector quantized visual words. Next, geometrical descriptions that are based on multiple small groups of points, *quads*, are introduced. Third, the extended method is applied to a realistic unconstrained dataset that is created for natural scene logo detection. The components of this template based detection method is explained in Section 4.2.2. In the light of the experiments presented in Section 4.2.3, it is observed that the proposed method provides a robust way to achieve template matching within datasets with harsh appearance variations. This method and its experimental evaluations has appeared in [48].

In Section 4.3, an evolved version of the method described in Section 4.2 is presented. This method, while inheriting most of the strengths of the previous, introduces a novel scheme for ameliorating the grouping problem in high clutter and large changes in scale. Detailed description of this method is introduced in Section 4.3.2. Additionally, this method is evaluated on a much larger dataset that is formed using data from another experimental dataset [152]. Experimental results presented in Section 4.3.3, supported the positive effect of the novel extension on the previous sections robust method, while generalizing the results to a more challenging dataset.

Algorithm 4.1 2D Affine Transform Estimation based Baseline Algorithm [75]

- thr_a: Appearance distance ratio threshold
- s_x : x difference bin size
- s_y : y difference bin size
- thr_e : Projection error threshold for estimated affine transform (Step C)
- N_c : Number of maximum iterations for transform verification (Step C)
- thr_c : Error change threshold for estimated affine transform (Step C)
- thr_r : Minimum size of a geometrically compatible group for recognition (Step D)

A. Appearance-based Potential Match Selection [75]

(1) Let spatial parameters location, scale and orientation (x, y, σ, θ) for i^{th} image feature be represented by f_{mi} and f_{qi} for model and query images, respectively.

(2) Let appearance description for i^{th} image feature be represented by d_{mi} and d_{qi} for model and query images, respectively.

(3) Let number of local features in model and query image be N_m and N_q, respectively.

(4) $M = \left\{(i, j) : j = \arg\min_k \|d_{mi} - d_{qk}\| \wedge \frac{\|d_{mi}-d_{qn}\|}{\|d_{mi}-d_{qj}\|} > thr_a,\ \forall n \in \left[1, N_q\right] \setminus j\right\}$

B. Transform Parameter Clustering using Hough Transform [142, 75]

(1) Let $f_{mr} = (x_{mr}, y_{mr}, \sigma_{mr}, \theta_{mr})$ and $f_{qs} = (x_{qs}, y_{qs}, \sigma_{qs}, \theta_{qs})$ represent the spatial parameters of matching feature pair $M_p = (r, s)$, where $p \in [1, N_M]$ and $N_M = |M|$.

(2) Let $V(i, j, k, l)$ be a four dimensional, sparse transform parameter histogram that quantizes the parameter space for x, y, σ and θ, respectively.

(3) $V(i, j, k, l) \leftarrow 0, \forall (i, j, k, l)$

(4) **for** $p = 1 \rightarrow N_M$ **do**

$\Delta x \leftarrow \frac{|x_{mr}-x_{qs}|}{s_x}$

$\Delta y \leftarrow \frac{|y_{mr}-y_{qs}|}{s_y}$

$\Delta\sigma \leftarrow \log_2 \frac{\sigma_{mr}}{\sigma_{qs}}$

$\Delta\theta \leftarrow \frac{((\theta_{mr}-\theta_{qs}) \mod 2\pi)\times 12}{2\pi}$

$V(i, j, k, l) \leftarrow V(i, j, k, l) + 1, \forall (i, j, k, l) \in Z_p$, where

$$Z_p = \{(a, b, c, d) : a \in \{\lfloor \Delta x \rfloor, \lceil \Delta x \rceil\} \wedge b \in \{\lfloor \Delta y \rfloor, \lceil \Delta y \rceil\} \wedge c \in \{\lfloor \Delta\sigma \rfloor, \lceil \Delta\sigma \rceil\} \wedge d \in \{\lfloor \Delta\theta \rfloor, \lceil \Delta\theta \rceil\}\}$$

end for

Algorithm 4.1 2D Affine Transform Estimation based Baseline Algorithm [75] (cont'd)

C. 2D Affine Transformation Estimation/Verification [75]

(1) Let 2D affine transformation (A) of a model point $[x\ y]^T$ to an image point $[u\ v]^T$ be modeled as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

(2) $S(i,j,k,l) \leftarrow 0, \forall (i,j,k,l), S'(i,j,k,l) \leftarrow 0, \forall (i,j,k,l)$

(3) **for all** $V(i,j,k,l) > 3$ **do**

Estimate $m_1, m_2, m_3, m_4, t_x, t_y$ using all pairs in M that voted for $V(i,j,k,l)$

for $count = 1 \rightarrow N_c$ **do**

Project all model points in M using estimated affine transformation A.

Calculate projection error (E_p) for all pairs in M.

$S(i,j,k,l) \leftarrow \left\{M_p : E_p < thr_e\right\}, M_p \in M$.

if $|S(i,j,k,l)| < 3 \vee \left(S(i,j,k,l) = S'(i,j,k,l) \wedge \left|E'_p - E_p\right| < thr_c\right)$ **then**

Break and process next $V(i,j,k,l)$.

end if

$S'(i,j,k,l) = S(i,j,k,l), E'_p = E_p$

Re-estimate $m_1, m_2, m_3, m_4, t_x, t_y$ using all pairs in $S(i,j,k,l)$.

end for

end for

D. Final Detection Decision [75]

(1) Merge and remove overlapping groups among $S(i,j,k,l)$.

(2) Declare positive detection if:

$$\exists (a,b,c,d) : |S(a,b,c,d)| > thr_r$$

⋆ *It is reported that the last stage of the baseline algorithm contains a probabilistic step for involving the total number of local features in the decision process. However, the provided implementation of the algorithm does not include this step.*

4.1 A Hybrid Method for Robust Correspondence Search Between Partially Planar Objects

A novel approach, which is based on combining the competence of interest point detectors to capture primitives and the capability of geometric constraints to discriminate between spatial configurations of these primitives is presented. In the proposed approach, the geometric constraints are enforced by means of barycentric coordinates (Section 3.3.1.1), a mathematical tool that has been utilized in the relevant literature as a neighborhood constraint. In the context of our research, however, these coordinates are utilized as a geometric description of a group of interest points. Using this description, local appearance descriptor based potential matches are intended to be filtered and evaluated according to their geometric consistency. This method is applicable to one-to-one image matching in its current form, and to classification tasks after extensions for appearance generalization. This work has been published in [47].

4.1.1 Motivation

Matching image regions between pairs of images is a common preliminary step in applications, such as 3D object modeling, object recognition, texture recognition and image retrieval [97, 153, 154, 155, 107, 156, 58, 157, 158]. The first strand of research in this area chooses regions to be matched by using segmentation, regular image grids or a randomized region location and scale selection method. The second strand of research [97, 153, 154, 155, 107, 156] focuses on detecting regions that fit into a predefined generic local pattern, namely interest points. There are many variations of interest point detection approaches [63, 64, 43, 159, 76], but all of them aim to locate covariant regions which can automatically adapt to the underlying image intensities in terms of location, scale and even affine shape. In this context, evaluation of interest point detectors [74, 44], along with various local descriptions to represent image regions [46] have also been presented.

Interest point detectors provide regions that are more repeatable with respect to their global and semi-global counterparts. On the other hand, since interest points are generally represented by relatively small regions, the local descriptions extracted from these regions have limited discriminative power. In addition, due to photometric and geometric transformations,

corresponding interest point regions might have significant descriptor distances. Under these circumstances, selecting the correct matches among many false counterparts, endures as an unsolved and challenging problem. In [75], a dynamic threshold is introduced in terms of the descriptor distance ratios. Matches filtered in this way then undergo a geometric consistency evaluation step, which involves quantization of the change in location, scale and orientation from one interest point region to its match in the other image. In the literature, there are alternatives to this approach, which consider the possibility of not being able to retrieve the correct matches of interest point regions as the nearest neighbor based on descriptor similarity as a common problematic situation [97, 153, 155, 107, 159].

The aforementioned methods have various solutions to interest point region matching that can be summarized roughly as follows: The methods in the first group [75] perform interest point matching based on local descriptor similarity leading to a strict decision before any other consistency check based on geometry. The second group of methods [155, 107], clusters interest point descriptors in training images and each interest point is allowed to match multiple clusters. Each match is assumed equally correct and contributes to statistical data that is fed to a higher level probabilistic model. The third class of approaches [97, 153] utilizes each potential match pair as a starting point for an iterative transform estimation and validation process and removes any inconsistent matches as part of this complex iteration process.

A preliminary step that can reduce the number of potential correspondences involves evaluating the potential match list of each interest point in terms of its geometric consistency with respect to its neighbors. Such a step should contribute to the solution of the intractable combinatorial problem of finding correspondences in the context of all of these approaches. This contribution would be twofold: First of all, descriptor-based similarity should then only be high enough to insert the right match to the potential match list. Secondly, the number and error ratio of correspondences input to any other higher-level analysis, such as probabilistic model construction or iterative transform estimation/validation, would be much lower.

In this section, we propose a method for filtering potential match lists based on coarse affine geometric consistency of a match with its neighbors. For this purpose, barycentric coordinates (Section 3.3.1.1) that are invariant under affine transformations are utilized. Presentation is organized as follows: In Section 4.1.2, we introduce our method (Algorithm 4.2) along with the approaches that inspired its development. Next, experimental results presented in Section

4.1.3 are followed by conclusive remarks in Section 4.1.4.

4.1.2 Components of the Method

In the proposed approach, geometric constraints are enforced by means of barycentric coordinates (Section 3.3.1.1). Using these coordinates as a geometric description, local appearance descriptor based potential matches are intended to be filtered and evaluated according to their geometric consistency. This method is applicable to one-to-one image matching by itself but can also be a preliminary part of training and classification phases of high-level inference frameworks. In this research, the method is explained through its use during one-to-one image matching. In this context, the term *query image* represents the image for whose interest points potential matches are searched in the *model image*.

4.1.2.1 Scale Invariant Interest Point Detection and Local Descriptor Extraction

There are two widespread categories of interest point detectors, namely corner and blob detectors (Section 2.1). Although, both categories are applicable in this preliminary step of the algorithm, a representative of the second category, SURF detector [76] is utilized for experiments that are presented in this paper.

In order to select potential matches for interest points, the proposed method relies on local descriptors. Among many alternatives (Section 2.2), the SIFT descriptor [75] is selected as a prominent one and utilized throughout the experiments.

4.1.2.2 Appearance Similarity-based Potential Match List Generation

Region descriptor of each interest point in the query image is compared to the interest points in the model image. After this comparison, a predefined number of interest points that have a descriptor distance smaller than a predefined threshold insert their interest point labels to the potential match list of the interest point under consideration. This step corresponds to Step A of Algorithm 4.2.

In the experiments, maximum length of potential match list is set to the value 5. The threshold here is determined by experimentation and depends on the utilized descriptor.

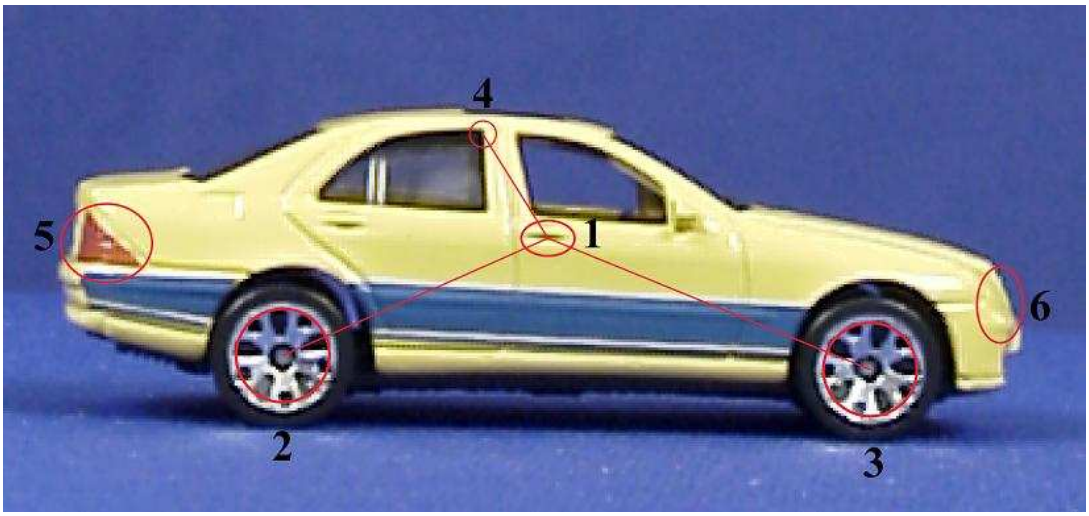


Figure 4.1: Example grouping according to euclidean distance. The neighbors of interest point 1 are $2, 3, 4, 5$ & 6. One of such triple combinations $(2, 3, 4)$ is illustrated.

4.1.2.3 Local Affine-Invariant Geometric Definition

Barycentric coordinates (Section 3.3.1.1) are computed for each interest point in terms of its neighbors. Each unique triple combination of neighbors (Figure 4.1) of an interest point along with its own coordinates generate a barycentric coordinate using Equation (3.26). This process is explained in Step B of Algorithm 4.2. This process should be considered as the geometric definition of each interest point by the relative positions of its neighbors. A sample partial result for the output of the process is given in Figure 4.2.

Both query and model image interest points are defined geometrically in terms of their neighbors, respectively. Model image geometric definitions in terms of interest point indexes and barycentric coordinates is called *Geometric Knowledge Base* for clarity in further references.

4.1.2.4 Extending Local Affine-Invariant Geometric Definitions of Query Image Interest Points using Potential Match Lists

At this point, two types of information are ready for the query image, namely the potential match list (Figure 4.3a) and the previously extracted barycentric coordinates for each interest point (Figure 4.3b).

In order to combine descriptor-based potential list and interest point based barycentric coordinates for the evaluation of potential matches in terms of geometry, interest point-based

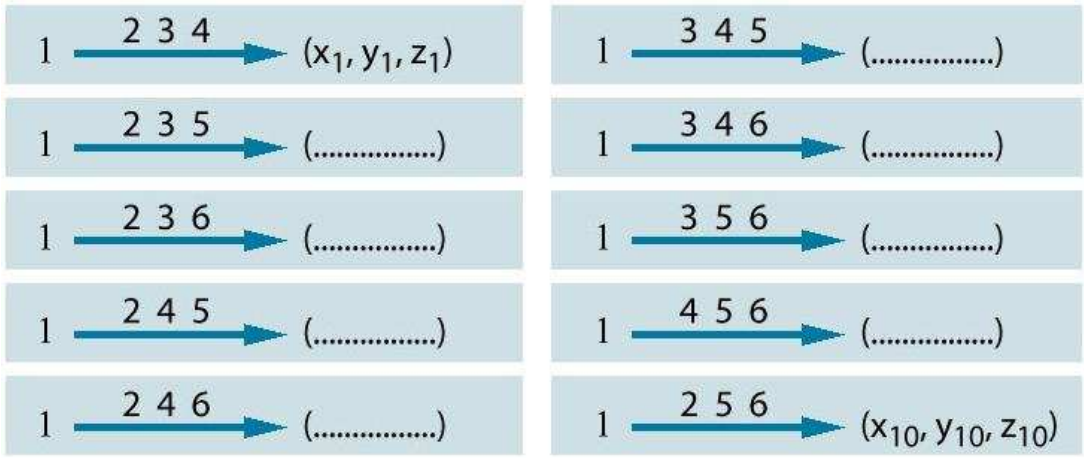


Figure 4.2: Example geometric definitions for a point and its neighbors. Each triple combination of neighbor interest points generate a barycentric coordinate for point 1. For the neighbors $2, 3, 4, 5$ & 6, all 10 triple combinations and coordinates are listed.

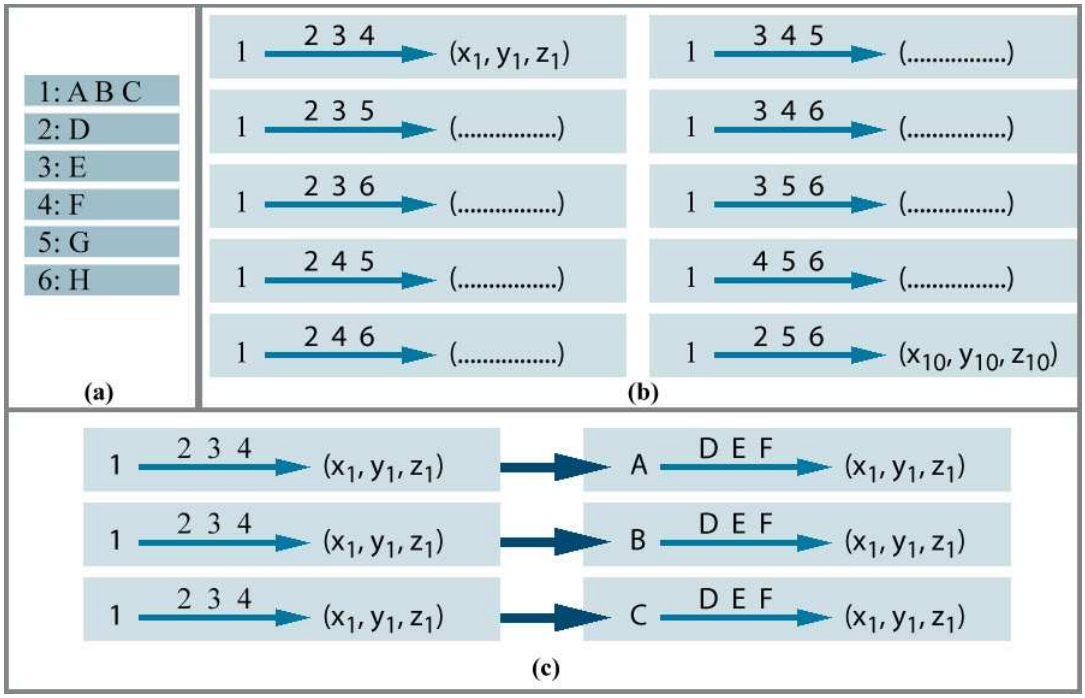


Figure 4.3: Joint matching of point groups in terms of geometry and appearance. For clarity, query interest points are represented by numbers $(1, 2, 3, \ldots)$, while model interest points are represented by capital letters $(A, B, C, \ldots)$. (a) Example potential match list, (b) interest point based barycentric coordinates, (c) Assignment based barycentric coordinates for the first triple combination in (b).

barycentric coordinates should be transformed into assignment-based barycentric coordinates. This step is performed for each possible assignment combination for each group of four neighboring interest points. The barycentric coordinates need not be recalculated, since they have already been calculated in the previous step. This process is explained in Step C of Algorithm 4.2. At the illustrative example of Figure 4.3, the center interest point (represented by 1) has 3 potential matches, while its neighbors each have one potential match. The resultant assignment based barycentric coordinates are given in Figure 4.3.c.

4.1.2.5 Accumulating Votes for Geometrically Consistent Groups

The output of the previous step is a set of barycentric coordinates in terms of potential matches, or in other words, assignments. This output is completely connected to the underlying query interest point indexes. This means that during data structure implementation, any four assignments (within model image interest point indexes) that generate a barycentric coordinate can be inversely mapped to the interest point indexes in the query image that gave rise to them. Each assignment based barycentric coordinate in the query image is now compared to the *Geometric Knowledge Base* of the model to find "consistent" geometric groups. Consistency is defined as follows:

- Overlapping neighbor interest points such as in $A \xrightarrow{BCD} (x_1, y_1, z_1)$ and $A \xrightarrow{BCD} (x_2, y_2, z_2)$
- Distance between Barycentric Coordinates (i.e. (x_1, y_1, z_1) and (x_2, y_2, z_2)) < Threshold (thr_b)

As an example, in Figure 4.3, each consistent group increases the votes of the underlying interest points forming it $(1, 2, 3, 4)$ to be assigned to model interest points (A, D, E, F) representing it. The voting process is illustrated in Figure 4.4, for these example interest point indexes. Illustrated process is explained in detail in Step D of Algorithm 4.2.

4.1.2.6 Vote-based Iterative Match Assignment

Iterative match assignment between query and model features is explained in Step E of Algorithm 4.2. Votes that are input to this step are accumulated in the previous step for a list

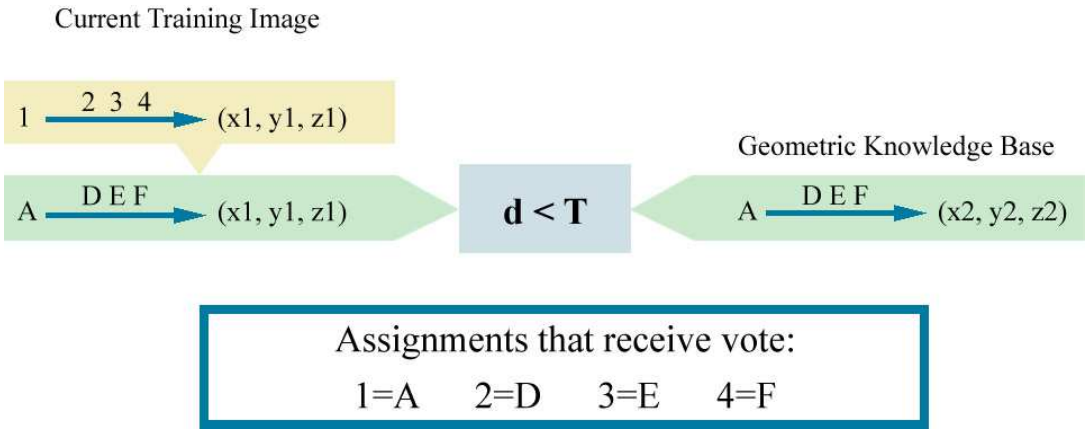


Figure 4.4: Voting process guided by joint description. Assignment-based barycentric coordinates consistent with geometric knowledge base cast votes.

of potential assignments which are generated earlier in the process (Section 4.1.2.2). An example vote list that will be used for following illustrative examples is given in Figure 4.5.

This input vote list is parsed to find the highest vote and the assignment with the highest vote (Figure 4.4, $1 = B$) is executed. Execution of an assignment triggers a chain reaction in the vote list and the assignment based barycentric coordinates list. All assignment-based coordinates that have voted for assignments conflicting with the executed assignment are invalidated (Figure 4.6). Votes cast by these invalidated groups are also invalidated. The invalidation is performed by decreasing the votes of the assignments induced by these groups (Figure 4.7).

The chain of triggered events concludes by the removal of all assignments of type $X = Y$ from the vote list, where X is the index of the query interest point that has just been assigned to a model interest point. Repeating the above steps, the vote list is processed iteratively until no query interest points that qualify for an assignment to a model interest point is left. An interest point can be left unassigned in three cases:

- Potential Match List is empty due to vast appearance difference with the descriptions in the model.
- None of its neighbor based barycentric coordinates are consistent with the Geometric Knowledge Base of the model image.
- None of the potential assignments of the interest point may have votes above the threshold.

Figure 4.5: Example result for the voting process

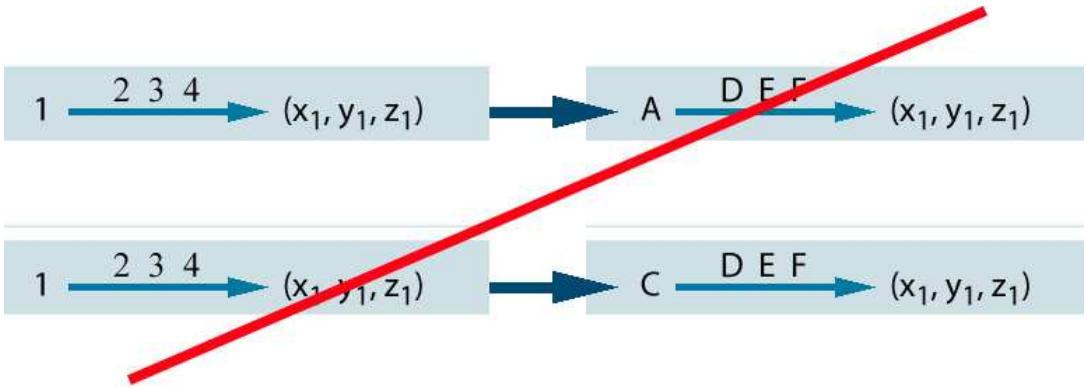


Figure 4.6: Examples of combinations that depend on assignments, which are invalidated due to execution of the assignment $1 = B$.

4.1.2.7 Vote Normalization and Filtering of Assigned Matches

The iterative nature of the assignment algorithm leads to a biased vote distribution in which interest points assigned in later steps lose more of their votes during the invalidation operations than the ones assigned in former steps. Therefore, for votes to become comparable, they should be normalized. This normalization is performed by reprocessing the assignments that are executed in later steps to let them invalidate the combinations inconsistent with them, but have cast vote for assignment executions before them. This forms the last step (Step F) of the method given in Algorithm 4.2.

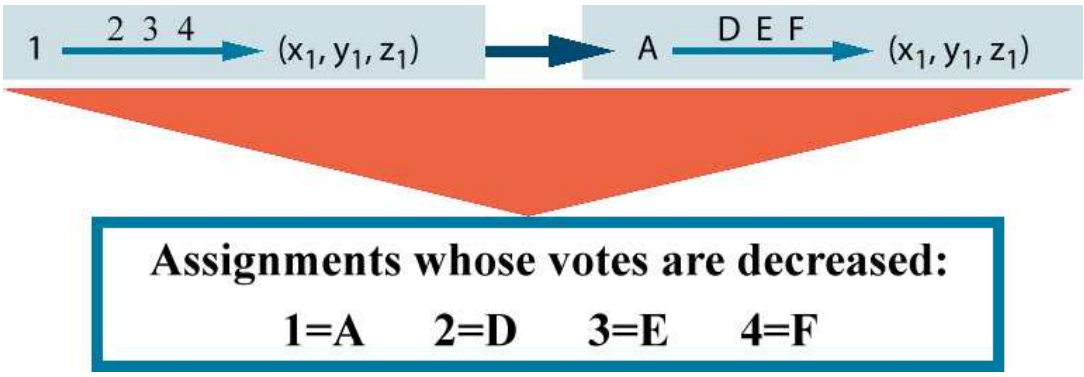


Figure 4.7: Votes decreased due to the invalidation of the first combination in Figure 4.6. The combination contains the assignment $1 = A$ that conflicts with the executed assignment $1 = B$.

Algorithm 4.2 Proposed Hybrid Robust Correspondence Search Method

- N_a: Appearance-based potential match list size
- N_s : Number of nearest spatial neighbors used for geometric description
- $g : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \longrightarrow \mathbb{R}^3$: Mapping from four image coordinates to Barycentric coordinates
- $d : \mathbb{R}^3 \times \mathbb{R}^3 \longrightarrow \mathbb{R}_+$: Mapping from two Barycentric coordinates to the distance between them
- thr_b : Barycentric distance threshold for compatibility

A. Appearance-based Potential Match List Population

(1) Let i^{th} image feature be represented by f_{mi} and f_{qi} for model and query images, respectively.

(2) Let number of local features in model and query image be N_m and N_q, respectively.

(3) Let Potential Match List $P_i = \{A, B, C, \ldots, \}$, where $P_i \subset \{f_{m1}, f_{m2}, \ldots, f_{mN_m}\}$, and members of P_i are the N_a nearest neighbors of f_{qi} in the model image in terms of appearance description.

B. Local Affine Invariant Geometric Definition For Query Images

(1) Let Nearest Neighbor List of f_{qi} be NN_{qi}, where $NN_{qi} \subset \left\{f_{q1}, f_{q2}, \ldots, f_{qN_q}\right\} \setminus f_{qi}$, and members of NN_{qi} are the N_s spatial nearest neighbors of f_{qi} in the query image in terms of L_2 distance.

(2) **for** $i = 1 \rightarrow N_q$ **do**

$C_{qi} \leftarrow$ All triplet combinations of elements in NN_{qi} $\left(\left|C_{qi}\right| = \binom{N_s}{3}\right)$.

for $j = 1 \rightarrow \binom{N_s}{3}$ **do**

$B_i(j) \leftarrow g(f_{qi}, C_{qi}(j))$

end for

end for

Step B is illustrated in Figures 4.1 & 4.2 by the following notational conventions:

- *Query features f_{qi} are represented by integer literals $i \in \left\{1, \ldots, N_q\right\}$.*
- *Barycentric coordinates $B_1(j)$ are represented by (x_j, y_j, z_j).*

Algorithm 4.2 Proposed Hybrid Robust Correspondence Search Method (cont'd)

C. Extending Local Affine-Invariant Geometry

(1) Appearance similarity information of Step A is utilized to extend the geometric definition in Step B to a *Geometric Knowledge Base* represented by Q as follows:

for $i = 1 \rightarrow N_q$ **do**

for $j = 1 \rightarrow \binom{N_s}{3}$ **do**

$E_{qi}(j) \leftarrow P_i \times P_a \times P_b \times P_c$, where P_i, P_a, P_b and P_c are potential match lists of query features f_{qi}, f_{qa}, f_{qb} and f_{qc} in the quadruplet $(f_{qi}, C_{qi}(j))$ defined in Step B.

Query feature indexes for f_{qi}, f_{qa}, f_{qb} and f_{qc} are also saved in each $E_{qi}(j)$ to be used in Step D.

$Q \leftarrow Q \cup E_{qi}(j) \times B_i(j)$, where $B_i(j)$ represents the quadruplet barycentric coordinates for j^th neighbor triplet $(C_{qi}(j))$ of query feature f_{qi}.

end for

end for

Step C is illustrated in Figure 4.3 with the following notational conventions:

- *Model features f_{mi} are represented by capital letters in the set $MF = \{A, B, C, \dots\}$, where $|MF| = N_m$.*
- *Query features f_{qi} are represented by integer literals $i \in \{1, \dots, N_q\}$.*
- *Barycentric coordinates $B_1(j)$ are represented by (x_j, y_j, z_j).*

Algorithm 4.2 Proposed Hybrid Robust Correspondence Search Method (cont'd)

D. Accumulating Votes for Geometrically Consistent Groups

(1) Model image geometric knowledge base constructed using Nearest Neighbor List NN_{mi}, nearest neighbor combinations C_{mi} and barycentric coordinates of quadruplets $(f_{mi}, C_{mi}(j)),\ \forall j \in \left\{1, \ldots, \binom{N_s}{3}\right\}$ in the model image is represented by M.

(2) Members of M are represented by $M_i = [(M_{iA}, M_{iB}, M_{iC}, M_{iD}), (x_{mi}, y_{mi}, z_{mi})]$, where $\{M_{iA}, M_{iB}, M_{iC}, M_{iD}\} \subset \{f_{m1}, f_{m2}, \ldots, f_{mN_m}\}$, and (x_{mi}, y_{mi}, z_{mi}) are barycentric coordinates of the quadruplet.

(3) Query image geometric knowledge base constructed using Nearest Neighbor List NN_{qi}, nearest neighbor combinations C_{qi} and barycentric coordinates of quadruplets $(f_{qi}, C_{qi}(j)),\ \forall j \in \left\{1, \ldots, \binom{N_s}{3}\right\}$ in the query image is represented by G.

(4) Members of Q are represented by $Q_i = \left[(Q_{iA}, Q_{iB}, Q_{iC}, Q_{iD}), (x_{qi}, y_{qi}, z_{qi})\right]$, where $\{Q_{iA}, Q_{iB}, Q_{iC}, Q_{iD}\} \subset \{f_{m1}, f_{m2}, \ldots, f_{mN_m}\}$, and (x_{qi}, y_{qi}, z_{qi}) are barycentric coordinates of the quadruplet.

(5) $V(q, m) \leftarrow 0,\ \forall q \in \left\{f_{q1}, f_{q2}, \ldots, f_{qN_q}\right\},\ \forall m \in \{f_{m1}, f_{m2}, \ldots, f_{mN_m}\}$

for all $Q_i \in Q$ **do**

$A_q \leftarrow (Q_{iA}, Q_{iB}, Q_{iC}, Q_{iD})\ ,\ G_q \leftarrow (x_{qi}, y_{qi}, z_{qi})$

$F \leftarrow (f_{qa}, f_{qb}, f_{qc}, f_{qd})$ represents query features that match to model features in A_q respectively.

for all $M_i \in M$ **do**

$A_m \leftarrow (M_{iA}, M_{iB}, M_{iC}, M_{iD})\ ,\ G_m \leftarrow (x_{mi}, y_{mi}, z_{mi})$

if $A_q = A_m$ **then**

$d_b \leftarrow d(G_q, G_m)$

if $d_b < thr_b$ **then**

$V(f_{qa}, Q_{iA}) \leftarrow V(f_q a, Q_{iA}) + 1\ ,\ V(f_{qb}, Q_{iB}) \leftarrow V(f_q b, Q_{iB}) + 1$

$V(f_{qc}, Q_{iC}) \leftarrow V(f_q c, Q_{iC}) + 1\ ,\ V(f_{qd}, Q_{iD}) \leftarrow V(f_q d, Q_{iD}) + 1$

end if

end if

end for

end for

$\hat{V} = \mid \{(a, b) : V(a, b) > 0\} \mid$

Algorithm 4.2 Proposed Hybrid Robust Correspondence Search Method (cont'd)

Step D is illustrated in Figure 4.4 with the following notational conventions:

- *Model features in model and query geometric knowledge base are represented by capital letters* $(A, B, C, \dots)$.
- *Barycentric coordinates of query image and model image are represented by* (x_1, y_1, z_1) *and* (x_2, y_2, z_2) *respectively.*

E. Vote-based Iterative Match Assignment

(1) $Q_{inv} \leftarrow \emptyset\ ,\ V_{exe} \leftarrow \emptyset$

while True **do**

$(\hat{q}, \hat{m}) \leftarrow \arg\max_{i \in \hat{V}} V(q, m)$

$vote \leftarrow V(\hat{q}, \hat{m})$

if $vote > 0$ **then**

$V_{exe} \leftarrow V_{exe} \cup (q_i, m_i)$

end if

for all $Q_i \in Q$ **do**

$A_q \leftarrow (Q_{iA}, Q_{iB}, Q_{iC}, Q_{iD})$

$P \leftarrow \Big\{(f_{qa}, Q_{iA}), (f_{qb}, Q_{iB}), (f_{qc}, Q_{iC}), (f_{qd}, Q_{iD})\Big\}$, where f_{qa}, f_{qb}, f_{qc} and f_{qd} are query features that match to model features in A_q, respectively.

$$F_{int} \leftarrow \{(q, m_1, m_2) : q \in \{q_1 : (q_1, m_1) \in P\} \cap \{q_2 : (q_2, m_2) \in V_{exe}\} \wedge (q, m_1) \in P \wedge (q, m_2) \in V_{exe}\}$$

for all $(q, m_1, m_2) in F_{int}$ **do**

if $m_1 \neq m_2$ **then**

$V(q, m_1) \leftarrow V(q, m_1) - 1$

$Q_{inv} \leftarrow Q_{inv} \cup Q_i$

end if

end for

end for

$Q \leftarrow Q \setminus Q_{inv}$

end while

Algorithm 4.2 Proposed Hybrid Robust Correspondence Search Method (cont'd)

Step E is illustrated in Figures 4.5, 4.6 and 4.7 with the following notational conventions:

- *Query features* f_{qa}, f_{qb}, f_{qc} *and* f_{qd} *are represented by integers.*
- *Model features in query geometric knowledge base are represented by capital letters* $(A, B, C, \dots)$.
- *Barycentric coordinates of query image are represented by* (x_1, y_1, z_1) *respectively.*

F. Vote Normalization and Filtering of Assigned Matches

(1) V is recalculated using the modified query geometric knowledge base Q.

(2) $\hat{V}_{exe} = \mid \{(a, b) : V(a, b) > 0\} \mid$

4.1.3 Evaluation: Multi-view Partially Planar Object Recognition in a Constrained Environment

The experimental results are presented for three specific analysis cases that are aimed to assess applicability of the algorithm under different conditions. The proposed method (Algorithm 4.2) is compared against Lowe's Hough Transform-based matching scheme, which is considered as a prominent baseline method. Lowe's method is already explained in detail in Algorithm 4.1. For each specific analysis case, quantitative performance evaluations are performed. In addition, quality of the results is presented via visual examples. Experiments are conducted on car image data from ETH-80 [160] and Pascal VOC 2007 [161] data sets. The number of images used in the performance comparison is limited due to the requirement of manual evaluation of the match results. Automatic interest point detection is performed by SURF [76] and local regions are represented by the SIFT descriptor [75].

4.1.3.1 Manually Selected Repeatable Locations

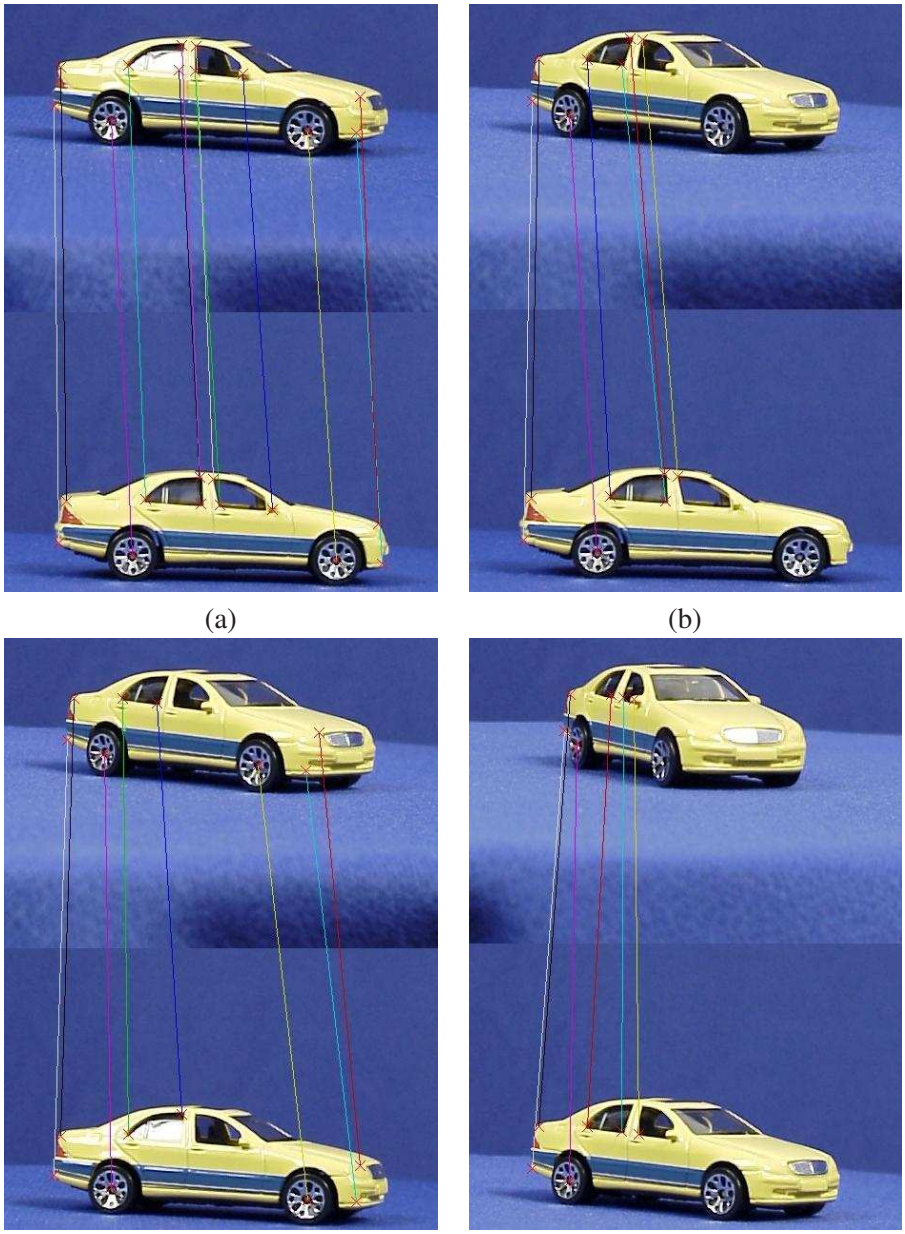
One of the most important problems about interest points is their varying re-detection performance. Since the proposed algorithm is applicable only for interest points that are reliably

Figure 4.8: Manually selected repeatable location

re-detected, a special analysis case is artificially created to isolate matching performance from repeatability performance of the interest point detector. In different views of the same object 12 feature locations are manually marked and at each of these locations, local scale is determined by LoG scale detector Section 2.1. Manually marked feature locations for original image is given in Figure 4.8. In this example, the proposed method is able to match all of the points accurately, while the baseline method can only match four of the possible twelve pairs. Quantitative comparison for matching results among 22°, 45°, 68° rotated and original views of the car object are presented in Table 4.2. Even though the interest point locations are accurately determined by manual intervention, descriptor distance harshly increases by view change. Therefore, the baseline approach considering only the nearest descriptor-wise neighbor fails in most of the cases. Matching results for four experimental pairs are given in Figures 4.9a-d.

4.1.3.2 Automatically Detected Interest Points in Different Views

Matching different views of the same object is the objective of this part of simulations, where rotated views (22-45 degree rotation in two different axis) of an object are matched to the original. Automatically detected interest points in the model image is given in Figure 4.10. Due to multiple visually identical appearances of visually identical parts (e.g. wheel), cross matches



(a) (b)

(c) (d)

Figure 4.9: Matches found by proposed algorithm for rotated pairs. (a) 22° rotated - Original pair, (b) 45° rotated - Original pair, (c) 45° rotated - 22° rotated, (d) 68° rotated - 45° rotated

Table 4.2: Number of accurate matches that are found in manually selected locations with the proposed method

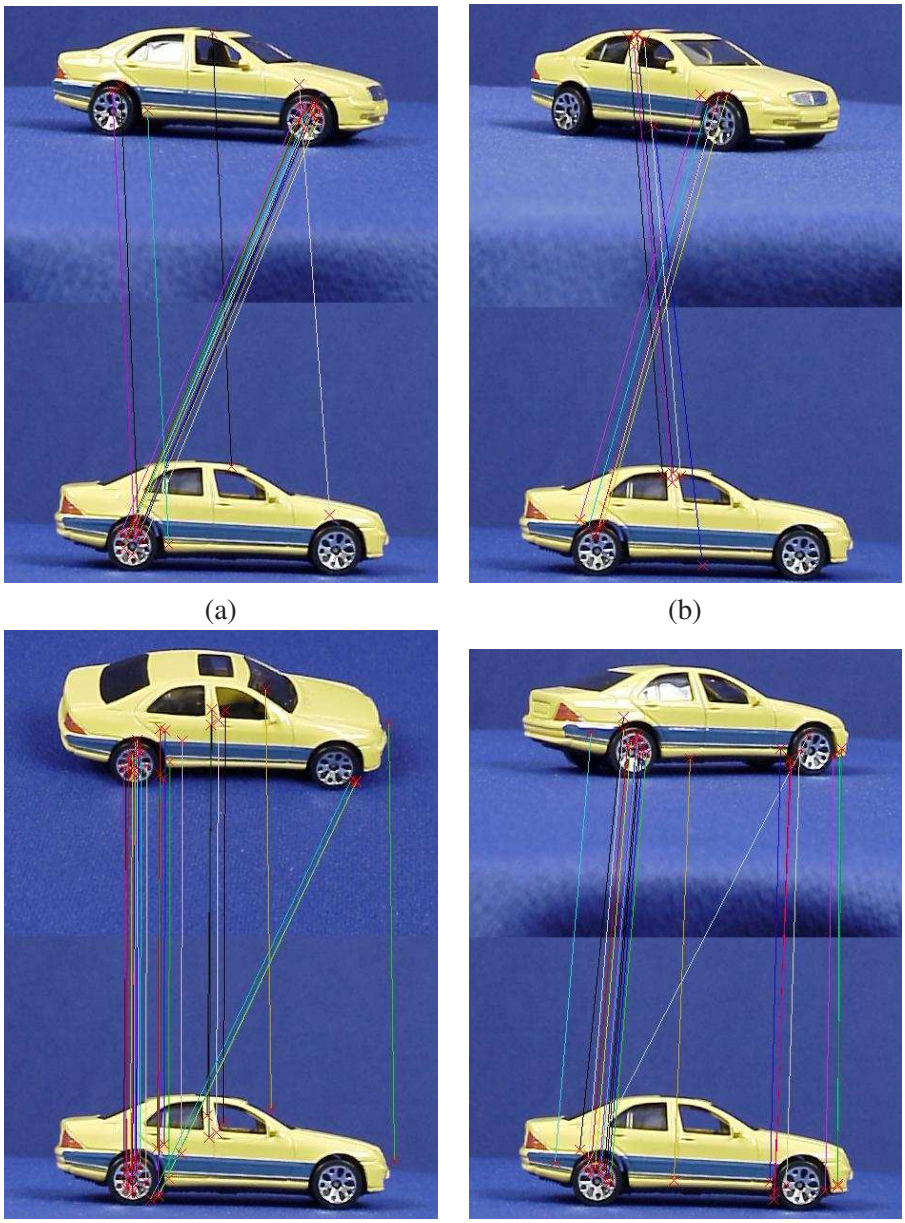
Matched Pairs (ETH-80 Car-1)	Number of Accurate (True) Matches Found	
	Proposed Method	Hough-based Method [75]
22° rotated - Original	12	4
45° rotated - Original	7	0
45° rotated - 22° rotated	8	0
68° rotated - 45° rotated	4	0

Figure 4.10: Automatically detected interest points of the model image

among the appearances of the same part are considered as accurate in performance evaluation. Quantitative results in Table 4.3 illustrate the superior performance of the proposed method when it is compared to the baseline method (Algorithm 4.1), in terms of finding correspondences between different views of the same object. Matching results for four experimental pairs are given in Figures 4.11.a-d.

4.1.3.3 Automatically Detected Interest Points in Images of Different Objects

In the last part of simulations, the matching performance of the proposed method under severe appearance variations is investigated. Although, the method is not specifically designed and enhanced for classification of semantically related objects (cars), promising results are obtained as it can be observed in Figure 4.12.a-d.



(a) (b)

(c) (d)

Figure 4.11: SURF detected interest point matches found by proposed algorithm for the rotated pairs. (a) 22° rotated (vertical) - Original pair, (b) 45° rotated (vertical) - Original pair, (c) 22° rotated (horizontal) - Original pair, (d) −22° rotated (vertical)- Original pair

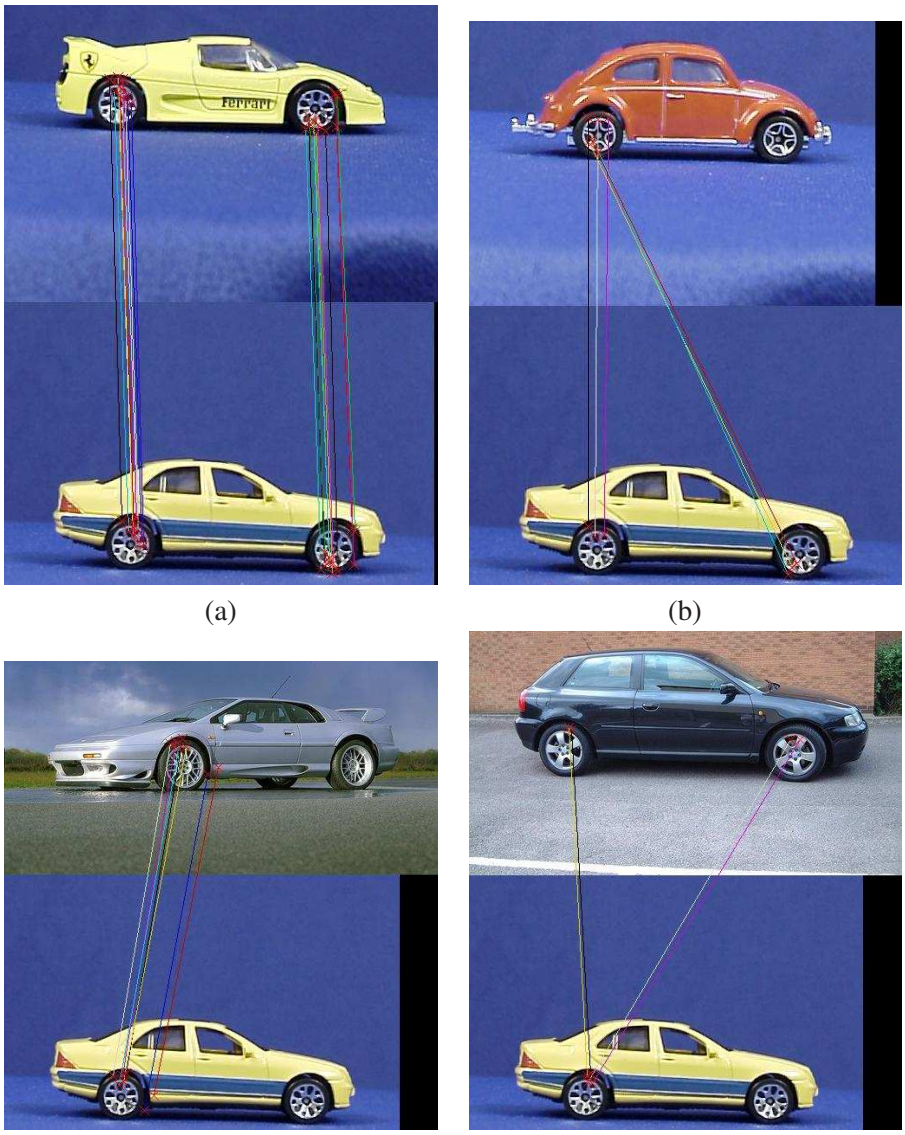


Figure 4.12: SURF detected interest point matches found by proposed algorithm for different objects. (a) ETH-80 Car6 - Car1 pair, (b) ETH-80 Car9 - Car1 pair, (c) Pascal 001576 - ETH-80 Car1 pair, (d) Pascal 003790 - ETH-80 Car1 pair

Table 4.3: Number of accurate matches that are found between different views of the same object

Matched Pairs (ETH-80 Car-1)	Number of Accurate (True) Matches Found	
	Proposed Method	Hough-based Method [75]
22° rot.(vert) - Original	15	12
45° rot.(vert) - Original	11	7
22° rot.(horz) - Original	29	15
−22° rot.(vert) - Original	21	18

Table 4.4: Number of accurate matches that are found between different car objects

Matched Pairs	Number of Accurate (True) Matches Found	
	Proposed Method	Hough-based Method [75]
ETH-80 Car6 - Car1	21	0
ETH-80 Car9 - Car1	6	0
Pascal 001576 - ETH-80 Car1	7	0
Pascal 003790 - ETH-80 Car1	4	0

These results suggest the applicability of the method for object classification tasks, after proper modifications for appearance and geometry generalization. Quantitative results and comparison to the baseline method is given in Table 4.4.

4.1.4 Conclusions

A method for detecting geometrically consistent interest point groups between different views of an object has been presented. Besides, the method has been tested on visually different but semantically and geometrically related objects. The experimental evaluations have shown the robustness of the method against appearance variations which severely degrade the performance of the baseline method (Algorithm 4.1). Therefore, the proposed method is more applicable for the tasks where geometrically consistent and reliable interest point groups are searched between different images of an object. On the other hand, due to naive neighborhood selection method (i.e. five nearest neighbors independent of an interest point scale-based dynamic constraint), for the test data many parts other than the wheels are lost. This result is due to the fact that in regions where interest points are dense, a neighborhood definition that is more advanced than N-nearest neighbors, should be utilized to cover larger potentially repeating configurations. This will be the next step in this research. Still, the performance

of the method on matching geometrically consistent parts of different semantically related objects (cars), unless it is affected by the neighborhood limitation, suggested its use in object classification tasks. Future work includes the enhancement of the method with appropriate generalization tools and its adaptation for the task of object classification.

4.2 A Framework for Joint Utilization of Vector Quantized Appearance and Geometry

A novel framework, involving the comparison of appearance and geometrical similarity of local patterns simultaneously via a combined description is presented. This method extends the preliminary work that is explained in Section 4.1 in various ways. First, the proposed description utilizes quantized appearance descriptors of interest points to avoid the necessity of matching each test descriptor to each template descriptor. Second, geometrical descriptions that are based on multiple small groups of points, quads, are introduced. This way, the geometric descriptions are based on multiple small groups of points, namely quads, instead of a single large group that is susceptible to partial transformations. Additionally, the extended method is applied to a realistic unconstrained dataset that is created for natural scene logo detection.

The introduced method renders one-to-many matching possible in contrast to its counterparts in the literature. Utilized geometrical descriptions are the same 2D affine transform invariants (Section 3.3.1.1) that are used in the previous method (Section 4.1). Its local nature and invariant based description utilization ability render the proposed algorithm robust to significant appearance changes, while being resistant to random false matches through simultaneous utilization of geometrical part of the descriptor. This generic, robust template matching technique is evaluated in an application of scene logo retrieval. The proposed algorithm proved itself in scene logo retrieval domain, where significant appearance changes, especially due to affine transformations take place [48].

4.2.1 Motivation

Detecting instances of a specific logo in images and video is of great importance for various applications. Logos can serve as an important cue for the presence of many semantic concepts,

such as political parties, companies and even illegal organizations.

In the literature, many methods have been proposed for logo detection [162, 163, 164, 165, 152]. However, the research has been mostly on the detection side within some constrained environments [162, 163, 164]. In contrast to logo detection in constrained environments, there are only a few algorithms proposed for the natural scene logo detection domain. An important research [165], which aims at detecting scene logos in frames of sport videos, utilizes edges, shapes and color composition. More recent work by Joly et al. [152] addresses the problem by using local interest point features. In their work, SIFT [75] descriptors extracted from template and test images are matched using L2 distance and then matches are filtered by a geometric consistency checking step.

In this paper, a novel approach involving the evaluation of appearance and geometry via a combined description is presented. This description utilizes quantized appearance descriptors of SIFT points to avoid comparing a test descriptor to all template descriptors in contrast with [152] and [75]. Geometrical descriptions are based on multiple small groups of points, namely *quads*, instead of a single large group. These advantages render the proposed algorithm robust to significant appearance changes, while being robust to random false matches through simultaneous utilization of geometrical description. The experimental results showing the robustness of the method by comparing it against a baseline algorithm (Algorithm 4.1) are provided in Section 4.2.3, which is followed by Conclusions Section (4.2.4).

4.2.2 Components of the Framework

In the proposed approach, appearance is represented by a codebook generated via clustering of local descriptions of interest point regions. Geometric constraints, on the other hand, are enforced on position information of interest points by means of barycentric coordinates (Section 3.3.1.1). Using these coordinates as a geometric description, quadruplet groups of codeword based potential matches are intended to be filtered and evaluated according to their geometric consistency. In this paper, we propose to solve the robust appearance representation problem by the help of clustered local descriptors, while filtering the side effect of decreased discriminative power using geometrical constraints. This way, we also avoid using all visual feature descriptions in an image and define images by codewords and their positions.

4.2.2.1 Scale Invariant Interest Point Detection and Local Descriptor Extraction

There are many examples of interest point detectors that can match to various distinctive parts of images (Section 2.1). DoG detector [75] is selected as a prominent one and utilized throughout the experiments, although it can be argued that other interest point detectors might also be exploited. It is noteworthy that the performance of some affine-invariant interest point detectors (Section 2.1) did not yield satisfactory results during simulations; hence, DoG is selected for the proposed algorithm.

Local feature description is performed by using SIFT [75]. This description is selected as a representative and prominent example of many local feature descriptor (Section 2.2).

4.2.2.2 Visual Codebook Creation and Usage

In order to avoid using all visual descriptions for the template and test images, and incorporate robustness to noise, an appearance codebook (or visual vocabulary) is obtained by using K-means algorithm [166]. The selection of a codebook representation has become widespread in image retrieval applications [167], due to its generalizing nature that allows efficient matching by capturing the variability of a particular feature type. Codeword assignment for each descriptor is performed by hard assignment of a single codeword. Appearance of each interest point used in the comparison is therefore represented by a codeword from the predefined codebook, instead of its original SIFT descriptor.

4.2.2.3 Local Affine-Invariant Geometric Definition by Barycentric Coordinates

Geometry of local feature groups are defined using the same geometric invariants, namely Barycentric coordinates. Barycentric coordinates are computed for each interest point in terms of its neighbors (Section 3.3.1.1). Each unique triple combination of neighbors (Figure 4.13) of an interest point generates a single barycentric coordinate. Throughout this section, a group of 4 interest points that are defined by a barycentric coordinate is denoted as *quad*. The extraction process of barycentric coordinates is explained in Section 4.1.2.3. This process should be considered as the geometric definition of each interest point by the relative positions of its neighbors. For each interest point, 10 spatially closest neighbors are used during the

Figure 4.13: Example grouping according to Euclidean distance. Neighbors of interest point 1 are $2, 3, 4, 5$ & 6. One of the triple combinations $(2, 3, 4)$ is illustrated.

experiments. A sample partial result for the output of the process has already been illustrated in Figure 4.2. An important detail here needs special treatment; any negative barycentric coordinate means that the center interest point is out of the triangular region defined by the positions of the selected neighbors. Therefore, quads with negative barycentric coordinates are considered invalid and not included in the retrieval.

4.2.2.4 Combined Description of Local Features using Appearance and Geometry

As already mentioned in the previous sections, visual descriptor of each interest point (i.e. SIFT) is transformed into a codeword and each quad formed around an interest point is defined geometrically via barycentric coordinates. These two representations are two independent dimensions of describing interest points, namely appearance and geometry.

In order to combine these two equally informative sources systematically, one needs to represent the components of quads by using their appearance. In our case appearance is represented by visual codewords. This operation is illustrated by an example in Figure 4.14. Interest points indexed by numbers 1 to 6, are each represented in terms of appearance by their respective codewords (Figure 4.14a). For this specific example, 1 is the center point and it has other point as neighbors around it. Each triplet of these neighbors is combined with the center point to form a quad (Figure 4.14b) and each of the interest points are represented by their codewords. As a result, each quad is represented by the codewords it is constructed from and the barycentric coordinates that define the positioning of these codewords. Combined representations of first and last quads in Figure 4.14b are given in Figure 4.14c. Two other properties, namely scale and dominant orientation of each interest point, are stored as additional information.

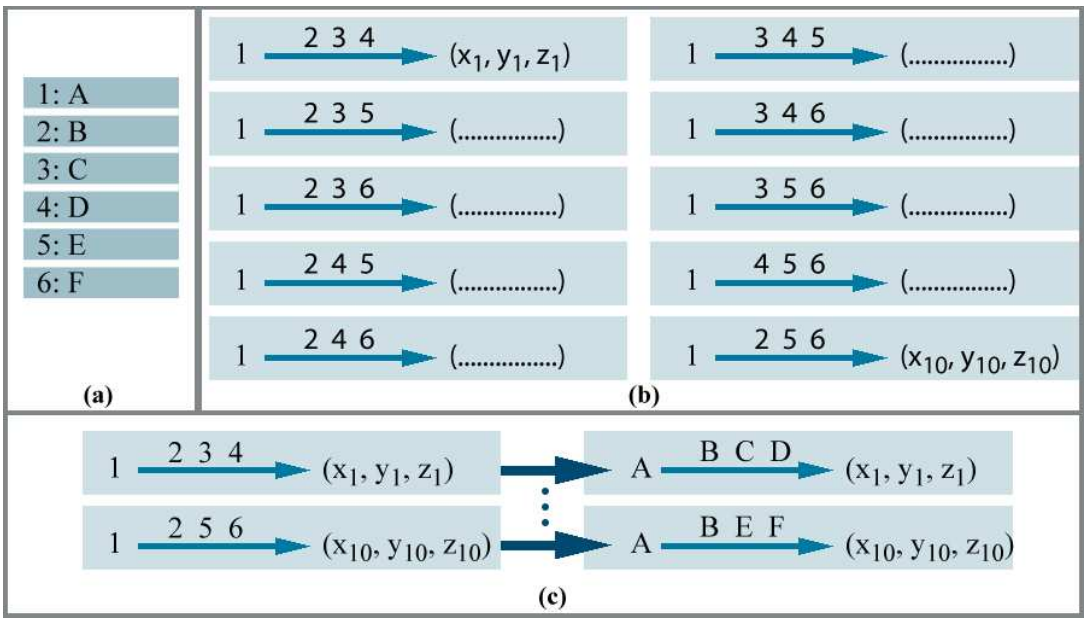


Figure 4.14: Joint representation of geometry and appearance of local feature groups: (a) Example codeword assignments; interest point 1 is assigned to codeword A, interest point 2 is assigned to codeword B, etc., (b) interest point based barycentric coordinates; *quad* $(1,2,3,4)$ has barycentric coordinates (x_1, y_1, z_1), *quad* $(1,2,5,6)$ has barycentric coordinates (x_{10}, y_{10}, z_{10}), etc., (c) Assignment-based barycentric coordinates for two *quads* in (b); *quad* $(1,2,3,4)$ corresponds to codeword-based *quad* (A,B,C,D) after assignment retaining the same barycentric coordinates, *quad* $(1,2,5,6)$ corresponds to codeword-based *quad* (A,B,E,F), etc.

Interest points for both template and test images undergo this combined definition process and comparison is based on these definitions. Combined definitions of template images in terms of codewords, barycentric coordinates and interest point spatial properties (i.e. scale, orientation) are together denoted as *Combined Visual Knowledge Base (CVKB)* for clarity in further references.

4.2.2.5 Comparison of Combined Visual Descriptions

The developed system searches patterns with a specific local appearance and geometry that are input to the system prior to retrieval process. These patterns are automatically extracted from the template images and converted into a database of patterns, i.e. CVKB. CVKB consists of information collected from representations of patterns that are observed in template images. The structure of each of these representations is given in Equation (4.1). Let CVKB consist of N quads q_i, $i \in 1,2,\cdots,N$. Each quad has four elements, i.e. interest points q_{ij}, with corresponding codewords cw_{ij}, scales σ_{ij}, and dominant orientations θ_{ij} for $j \in 1,...,4$. Barycentric

coordinates of each quad q_i are represented by b_{ik}, where $i \in 1, \cdots, N$ and $k \in 1, \cdots, 3$.

$$q_i : (c\vec{w}_i, \vec{\sigma}_i, \vec{\theta}_i, \vec{b}_i) \tag{4.1}$$

Quads within test images are compared with the CVKB by this representation. The comparison is performed for each field of the representation separately. Let q_t be the test quad to be compared to CVKB. The first metric to be compared is the codewords of the two quads, if they do not match then there is no need to compare other fields. If the codewords match then remaining three fields are compared. Let q_i be defined as in Equation (4.1) and q_t be defined by $(c\vec{w}_t, \vec{\sigma}_t, \vec{\theta}_t, \vec{b}_t)$. Compatibility of these two quads is assessed as follows:

1. If $\forall j \in 1, \cdots, 4, cw_{ij} = cw_{tj}$, then continue; else quads are incompatible.
2. Compute $\Delta\sigma = (\Delta\sigma_1, \Delta\sigma_2, \Delta\sigma_3, \Delta\sigma_4)$ and $\Delta\theta = (\Delta\theta_1, \Delta\theta_2, \Delta\theta_3, \Delta\theta_4)$, where,

$$\Delta\sigma_j = \left\lfloor \log_2 \frac{\sigma_{tj}}{\sigma_{ij}} \right\rfloor \tag{4.2}$$

$$\Delta\theta_j = \left\lfloor \frac{\left(\left(\theta_{tj} - \theta_{ij}\right) \mod 2\pi\right) \times 12}{2\pi} \right\rfloor \tag{4.3}$$

3. Compute barycentric coordinate distance d_b by using Equation (3.28).
4. q_i is compatible with q_t, if and only if:
   (a) $d_b < thr_b$
   (b) $\max(\Delta\sigma_j) - \min(\Delta\sigma_j) < thr_\sigma$
   (c) $\max(\Delta\theta_j) - \min(\Delta\theta_j) < thr_\theta$

$\Delta\sigma$ is the quantized scale ratio vector between two quads. Quantization is performed in $\log 2$ scale. Similarly $\Delta\theta$ is the quantized orientation difference vector with a bin size of $\pi/6$ radians. These three threshold parameters are used to precisely control the geometrical consistency for different metrics. The first parameter, which is probably the most dominant among all, is the threshold on barycentric distances, thr_b. This parameter controls the overall geometrical consistency of two quads and is applied on the distance values directly without quantization. The second and third threshold parameters are applied on coarsely quantized difference values. The second parameter, thr_σ, is used to limit the heterogeneity in scale

Figure 4.15: Sample positive and negative images from the dataset are given in first and second rows respectively.

change due to the affine transformation that part of the logo undergoes. The third parameter, thr_θ, provides the means to limit the heterogeneity of change in local orientation of patches that construct a quad.

4.2.3 Evaluation: Joint Utilization of Appearance and Geometry for Scene Logo Retrieval

In order to evaluate the performance of the proposed algorithm, A well-known and frequently encountered brand logo is selected as a typical example (Coca-Cola). From many images automatically downloaded as a result of Google queries, "coca cola billboard", "coca cola truck", "coca cola car", "coca cola truck", "coca cola building", "coca cola table", only 136 with significant amount of geometrical and appearance distortions were selected as the positive set subjectively. Next, 150 negative images are selected from the images that were returned as the result of the query "logo" and "billboard". The aim of these selections were to test the algorithms on data which is realistic in terms of both positive and negative samples (Figure 4.15). Two of the collected brand logos are selected as template images (Figure 4.16).

The visual codebook created for assigning appearance descriptors of test and template images consists of 128 codewords, and it is constructed from a different dataset of images that contain 5000 randomly selected retrieved images. 50% of these images are downloaded from web queries by using "logo" and the other 50% using "photo" keyword. During the experiments, constraints on barycentric distance, scale changes and orientation changes, that are explained in the previous section are applied and their performances are measured separately. Next, performance of these constraints in combination are observed.

In addition to the previously explained parameters, a straightforward extension to the template images is also evaluated. This extension, which is denoted as Artificial Template Extension (ATE) throughout the text, consists of artificially applying a number of (six for these

Figure 4.16: Template logos that are selected to be used as query in the experiments.

experiments) viewpoint changes and some (four during these simulations) scale transforms on template images. These newly created images are then used in the same manner as the original template images. Performance obtained by parameters at different values are measured in terms of true positive and false positive rates. Any match to quads of any of the template images are counted as a positive. The result of these measurements are presented via Receiver-Operating Characteristic (ROC) curves. At the end, the performance of the proposed algorithm is compared against a baseline algorithm, that is prominent in the literature. The baseline algorithm is selected as the most recommended matching method for SIFT [75]. This algorithm also has a geometric compatibility checking stage to compensate for affine changes, and due to its nature directly compares SIFT descriptors without quantization. This method is explained in detail in Algorithm 4.1. There are two parameters that are varied to analyze the performance of the baseline algorithm. The first of these parameters is the ratio of first nearest matches descriptor distance (thr_a) to the second. Any match having a ratio higher than the threshold is rejected. Three threshold values, $0.6, 0.7$ and 0.8 are used during the experiments. The other parameter used to evaluate the performance of the baseline algorithm is a lower limit on the number of matches (thr_r) that survived at the end of the geometric compatibility process. A much wider range of $[0, 250]$ is used for this parameter.

During the first part of the experiments, three constraints, namely barycentric distance, scale change heterogeneity and orientation change heterogeneity are applied in isolation by using threshold parameters thr_b, thr_σ and thr_θ, respectively. This is performed by changing one

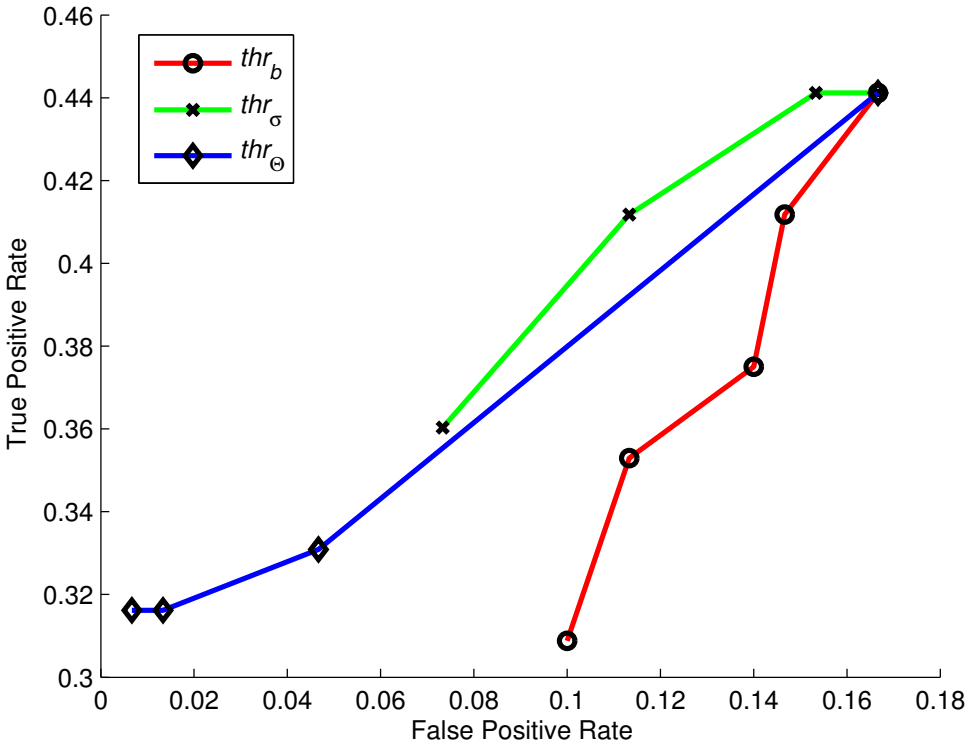


Figure 4.17: Effect of thr_b, thr_σ and thr_θ parameters on performance in isolation for single template case for the proposed algorithm.

of the threshold parameters in its predefined range (Table 4.5) and relaxing the other two in order to remove their influence on the performance. Parameters thr_σ and thr_θ are set to infinity, when relaxed. Effects of these parameters on performance are presented both for single template (Figure 4.17) and ATE cases (Figure 4.18).

In the second part of the experiments, parameters, whose effects were analyzed in isolation in the previous part, are combined. Performance of the proposed algorithm considering the joint effect of the parameters thr_b, thr_σ and thr_θ are compared against the baseline algorithm (Algorithm 4.1) for both single template and ATE cases (Figure 4.19). These comparisons are presented via ROC curves that are obtained by searching the highest possible true positive rate for various limits on the false positive rate. The parameter combinations that lead to highest true positive rates for some informative points of the ROC curve are presented in a separate figure (Figure 4.20). Representative successful recognition instances for the algorithm are given in Figure 4.21. Similarly, some representative cases where the proposed algorithm fails are provided in Figure 4.22.

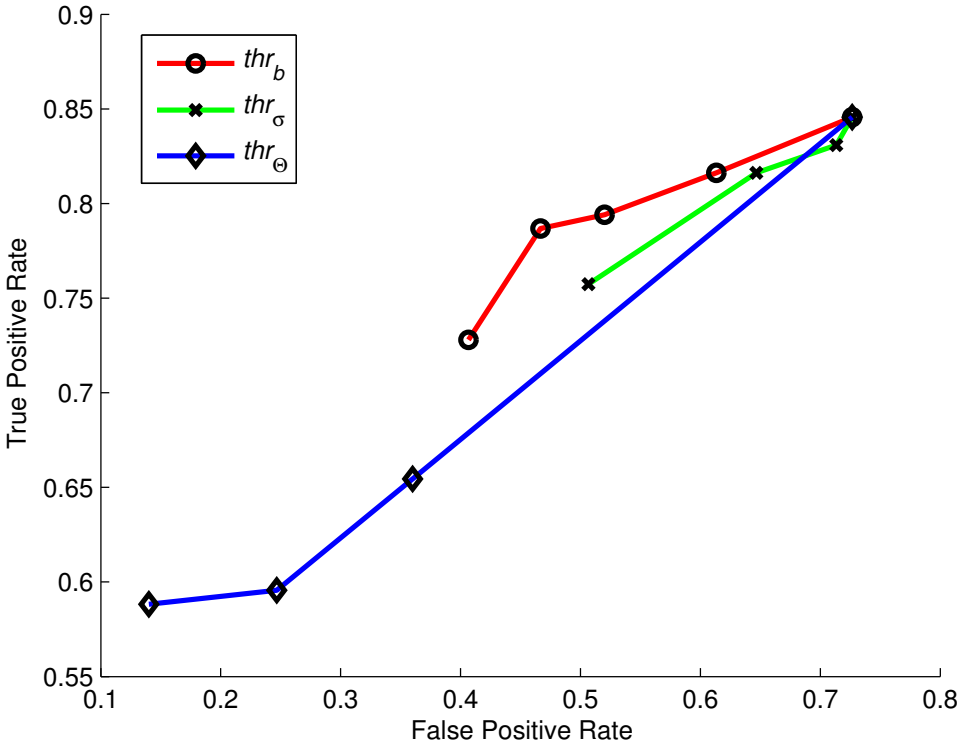


Figure 4.18: Effect of thr_b, thr_σ and thr_θ parameters on performance in isolation for ATE case for the proposed algorithm.

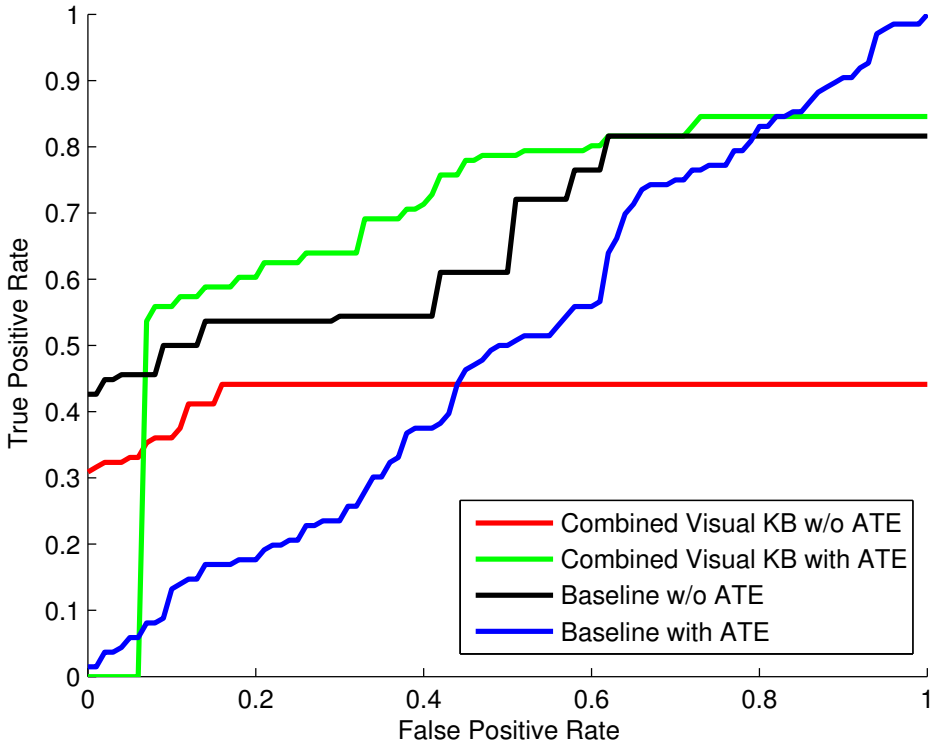


Figure 4.19: Performance comparison of the proposed algorithm, i.e. Combined Visual Knowledge Base (KB) with the baseline (Algorithm 4.1) for single template and ATE cases.

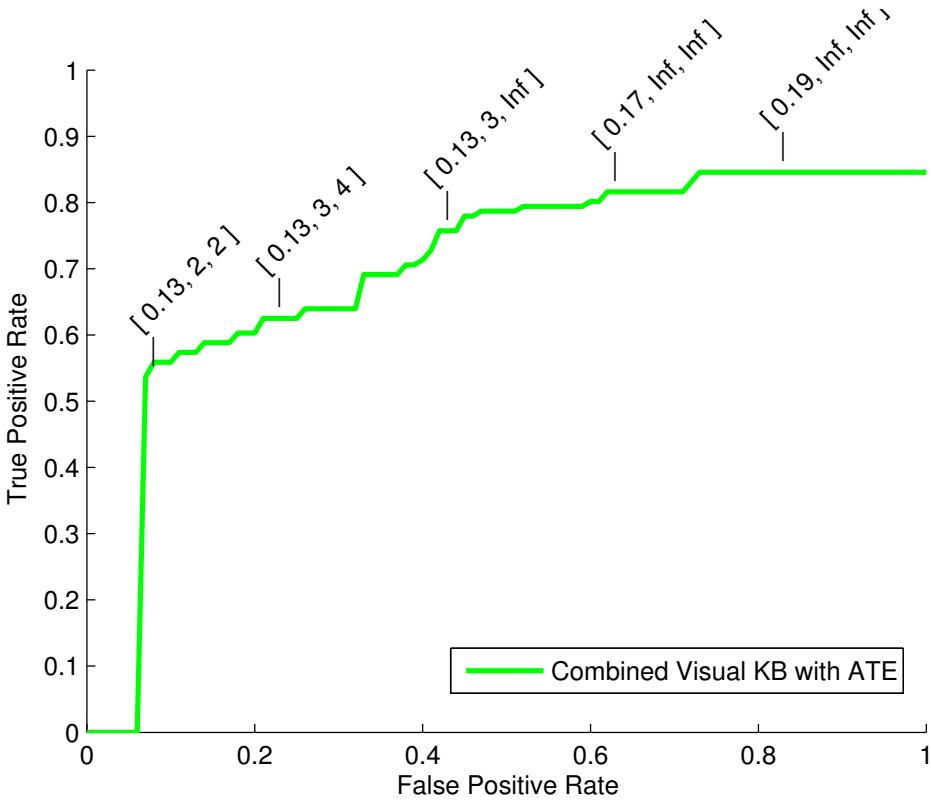


Figure 4.20: Performance of the proposed algorithm, i.e. Combined Visual Knowledge Base (KB) with ATE. Parameters, $(thr_b, thr_\sigma, thr_\theta)$ that lead to the results shown are displayed on the curve at informative points.

Table 4.5: Simulation parameters for the proposed and the baseline (Algorithm 4.1) methods

	Parameters	Min. Value	Max. Value	Step Size
Proposed	thr_b	0.11	0.19	0.02
	thr_σ	2	4	1
	thr_θ	2	4	1
Baseline	Distance Ratio	0.6	0.8	0.1
	# Pt. Matches	0	250	1

Some points need to be clarified in order to ensure accurate interpretation of Figure 4.19 and 4.20. First of all, the steady start of the proposed algorithm with ATE until a non-zero false positive rate is due to some negative images that are falsely retrieved in all parameter combinations. Next, there are also steady regions at other parts of the curves, especially towards high false positive rate. This result is due to the fact that with the parameter ranges defined for our experiments, it is not possible to achieve the retrieval of some positive data. As previously explained, the curves show the highest true positive rate with respect to upper limit on the false positive rate. Therefore, if true positive rate does not increase, as one increases the upper limit on false positive rate, then horizontal steady regions are observed on the curve.

Experiments performed also include utilization of the stop-list approach [168] that is frequently applied in text retrieval. In a separate strand of experiments, very frequent codewords and quads are detected and dismissed before applying the proposed algorithm. Despite its success in text retrieval domain, this approach does not enhance the performance of the proposed algorithm in this domain based on the conducted experiments. Therefore, the experimental results related to these experiments are not included in this text. Sample results showing the strength of the proposed method and respective interest point matches are given in Figure 4.21. The representative colors of the interest points (other than yellow) are deliberately selected to aid in identifying pairs of quad centers that are matched by the algorithm. The interest points that are shown with yellow color are neighbors of the centers of the matched quads.

Figure 4.21: Examples of successful recognition results. Template interest points are shown in the upper row, test interest points and images are given below them.

4.2.4 Conclusions

A template-based matching approach for matching distinctive groups of interest points has been presented. In this approach, robustness of appearance representation is enhanced by using clustered local descriptors, while compensating the side-effect of the diminishing individual discriminative power by the help of group-based geometrical constraints. The first part of the experiments presents the individual effect of each of the three constraints that constitute the geometrical power of the approach. In these experiments, it is observed that each of these constraints have similar effects on filtering false positives and none of them can individually be considered superior to the others (Figure 4.17). On the other hand, for the ATE case, it can be observed that each of these parameters prove robust against a radical increase in the number of templates (Figure 4.18). It can be seen that utilizing any of these parameters, increase in false positive rate can be successfully controlled, while increasing the true positive rate significantly.

The results obtained from individual assessment of the parameters of the proposed algorithm led us repeating the experiments to assess combined effects of the parameters. The results of these experiments have demonstrated that these parameters reinforce each other's perfor-

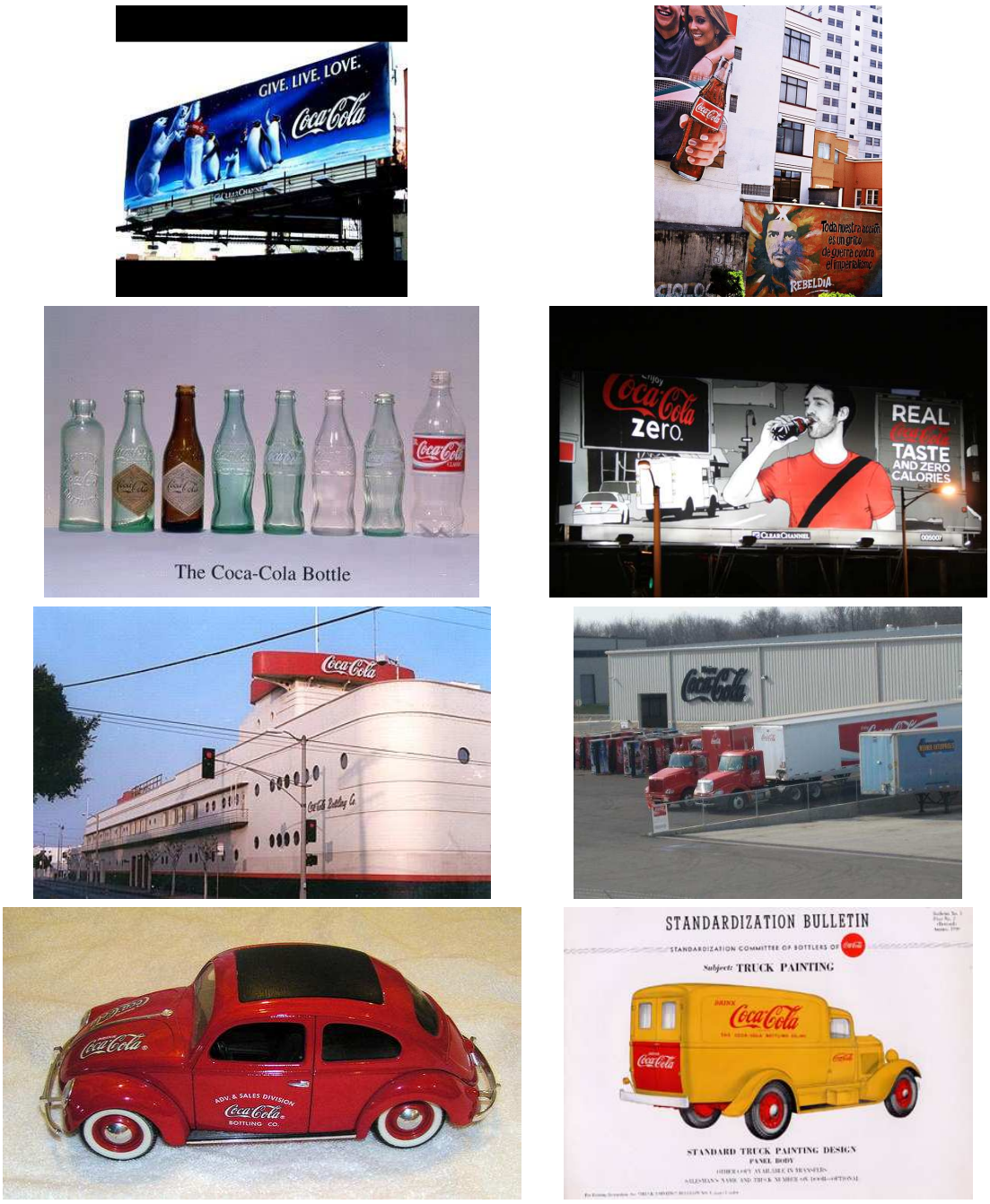


Figure 4.22: Examples of cases where recognition fails. In most of the cases, logo region undergoes harsh photometric effects.

mance in filtering false positives, and especially for the ATE case profited significantly from the increase in the number of templates (Figure 4.19). The baseline method (Algorithm 4.1), on the other hand, could not profit from the increase in the number of templates and since it does not limit the increase in false positive rate, even with a reasonable parameter search. This is coherent with the previous research [152], which aim to model the false positive distributions and properties, in order to profit from larger template sets.

As an additional benefit, these experiments helped us gain insights about the nature of transformations that planar logos undergo in real life. The parameters that led to various performance results (Figure 4.20) have shown that support region size or scale of interest points that lie close together can vary significantly (threshold thr_σ needs to be increased to at least 2 -meaning a scale ratio of 4- in order to obtain a true positive rate of around 0.55) . In addition, affine distortion that govern the geometrical change in a planar logos, also changes the local description of the interest point regions significantly. This fact can be directly observed from the heterogeneity of dominant orientation change in positive images (threshold thr_θ needs to be increased to at least 2 -meaning an orientation difference of $\pi/3$ radians- in order to obtain a true positive rate of around 0.55).

In the cases where ATE is not applied, the performance of the proposed algorithm is lower than the baseline algorithm. This is considered as a direct consequence of the fact that repeatable neighborhoods are an integral part of the proposed approach. Therefore, scale change that changes the neighborhood of local features by increasing or decreasing the total number of interest points significantly, degrades the performance of the CVKB method. For the time being, ATE seems to ease this effect, but this seems to be the soft spot of the method and need to be addressed in the future. The proposed method, in the light of the experiments, provides a robust way to achieve template matching with large template sets and using much less discriminative yet repeatable local features. The approach is open to improvements, such as generalization on template generation step, using the common repeatable quads among the template images and generalizing them. After this kind of modifications, the approach can be adapted to classification and other high-level problems as an intermediate layer.

4.3 An Extended Framework for Joint Utilization of Vector Quantized Appearance, Geometry and Significance-based Grouping

The experimental results obtained and the conclusions that are drawn led to the development of an evolved version of the method described in Section 4.2. This method inherits the strengths of the previous method, which are the utilization of generalized local appearance descriptors and semi-local geometric definitions based on geometric invariants. As stated in Section 4.2.4, despite the utilization of scale invariant local descriptions and affine invariant geometric descriptors, the invariance of the method against scale change as a whole is threatened under significant scale changes. In addition, presence of extreme clutter creates a similar effect on the neighborhoods of local features. In order to be able to select the same local feature groups that form the quads, it is necessary to introduce a grouping constraint that is robust to scale change and extreme clutter. In this section, an evolved method that introduces a novel scheme for ameliorating the grouping problem in high clutter and large changes in scale. Detailed description of this method is introduced in Section 4.3.2. Additionally, this method is evaluated on a much larger dataset that is formed using data from another experimental dataset [152]. Experimental results presented in Section 4.3.3, supported the positive effect of the novel extension on the previous method (4.2) [49].

4.3.1 Motivation

Instance-level recognition of a specific logo in images and video is of great importance for various applications. The areas that mostly benefit from the statistics of the appearance of logos are advertisement, marketing and sponsoring sectors. Sponsors can project the effect of their sponsoring contracts by measuring the appearance frequency of their logos. Another application of logo detection is information retrieval in large image and video databases.

In the literature, many methods have been proposed for logo detection [169, 170, 162, 163, 171, 172, 173, 174, 164, 165, 152]. However, the research has been mostly on the detection side within some constrained environments. The most constrained case is detection and classification in text documents [169, 170, 162], where logos are located on a homogeneous background, with no occlusions, or 3D transformations. Another common application is TV channel logo detection [163, 171, 172, 173, 174, 164], whereas in this domain, logos are lo-

cated on heterogeneous backgrounds and most of the time subject to only minor appearance changes related to broadcast quality. The methods in this domain have the advantage of using the temporal dimension in addition to the 2D image frames. Some of these methods handle the cases, in which the logos being searched are transparent [173, 174] or even animated [164]. On the other hand, none of these methods are designed to cope with natural and realistic scenes, in which the logos are an integral part of the scene. This realistic case needs a special treatment, such as handling occlusions, 3D transformations, and radical appearance changes, which is beyond the reach of the methods that have been discussed until now.

In contrast to logo detection in constrained environments, there are only a few algorithms proposed for the natural scene logo detection domain. An important research [165], which aims at detecting scene logos in frames of sport videos, utilizes edges, shapes and color composition. More recent work by Joly et al [152] addresses the problem by using local interest point features. In this work, SIFT [75] descriptors that are extracted from template and test images are matched using L_2 distance and then these matches are filtered by a geometric consistency checking step. This last step is based on iterative estimation of an affine transformation from a group of points that are already matched using appearance description, and filtering of inconsistent one from the matched point list. This approach considers matching appearance and geometry as two different steps, and uses plain SIFT descriptors, which are based on small local patches that are not discriminative enough on their own, for matching as the first step. Only after the next step of iterative transform estimation, these matches are filtered according to their geometrical consistency as a group. In the first step, since local descriptions are not discriminative enough, and only the nearest L2 neighbors in terms of the SIFT descriptor are eligible for the geometric consistency check, false SIFT matches might easily eliminate the true correspondences. In addition, estimating a single transformation for an image is not enough for many cases. Since logos are frequently printed on non-rigid or non-flat supports different parts of them may undergo different transformations.

For addressing these challenges, a novel approach involving the evaluation of appearance and geometry via a combined description is presented in this paper. This description utilizes quantized appearance descriptors of SIFT points to avoid matching each test descriptor to each template descriptor. In order to filter out keypoints that do not belong to any meaningful matching group, as a preliminary step, a probabilistic approach is utilized. Proposed method renders one-to-many matching possible in contrast to [152] and [75]. Geometrical

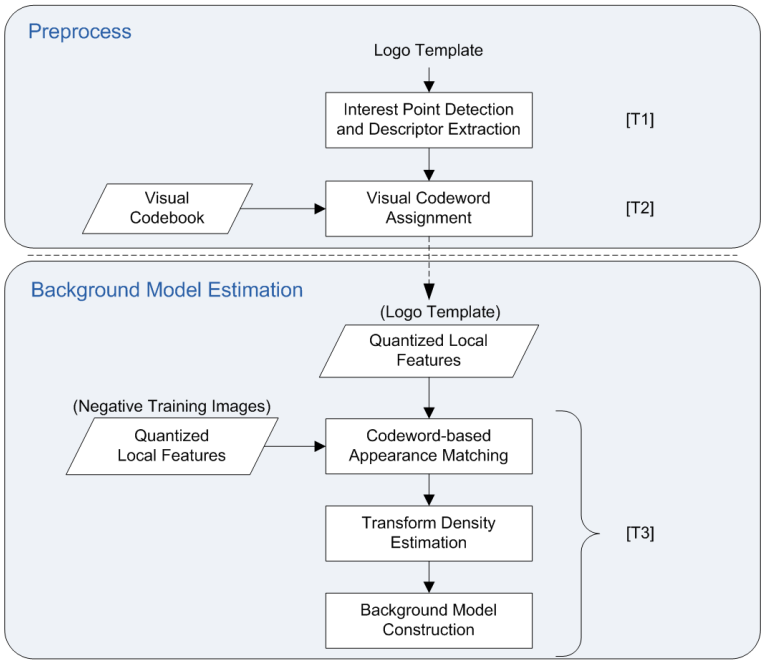
descriptions of keypoints that belong to promising match groups are based on multiple small groups of points, *quads*, instead of a single large group. These advantages render the proposed algorithm robust to significant appearance changes, while being robust to random false matches through simultaneous utilization of geometrical description. This method, which has its roots at an earlier approach [47], is explained in detail in Section 4.3.2. The experimental results showing the robustness of the method by comparing it against a baseline algorithm (Algorithm 4.1) is given in Section 4.3.3. In the last section, Section 4.3.4, experimental evidence is combined with some remarks in order to reach some very important conclusions and identify possible research directions.

4.3.2 Components of the Framework

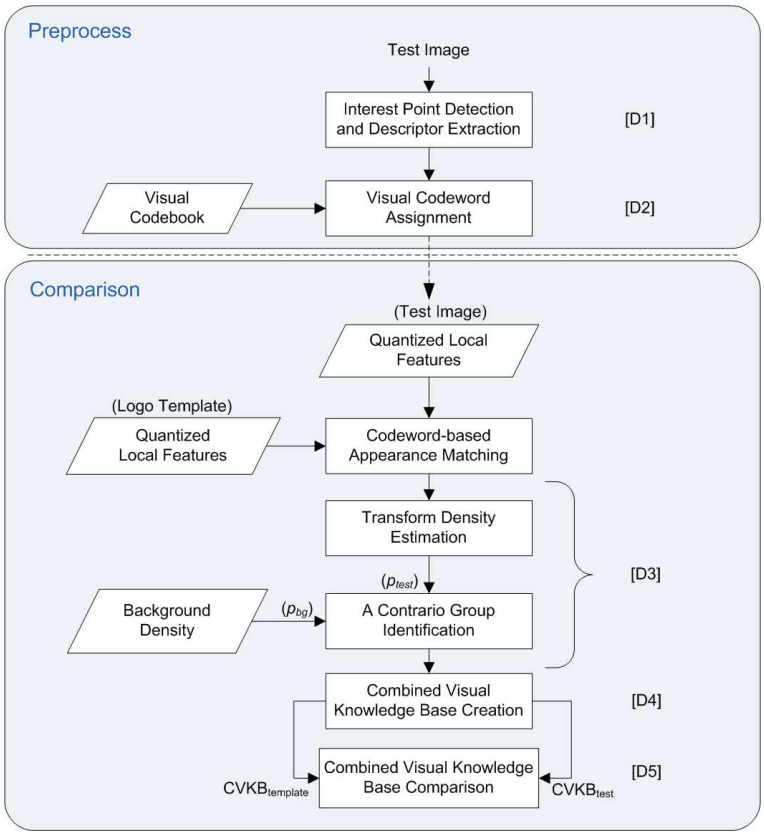
In the proposed approach, appearance is represented by a codebook generated via clustering of local descriptions of interest point regions. Geometric constraints, on the other hand, are enforced on the position information of interest points by means of barycentric coordinates (Section 3.3.1.1). Using these coordinates as a geometric description, quadruplet groups of codeword based potential matches are intended to be filtered and evaluated according to their geometric consistency. In this paper, we propose to solve the robust appearance representation problem by the help of clustered local descriptors, while filtering the side effect of decreased discriminative power using geometrical constraints. In this way, we also avoid using all visual feature descriptions in an image and define images by codewords and their positions.

An overview of the proposed method is given in Figure 4.23. Training and detection stages are illustrated in two separate flowcharts. Training stage (Figure 4.23a) consists of two main parts. The first one is the preprocess of extracting local features and then quantizing their descriptors using a codebook. This preprocess is applied to both logo templates and negative training. Next, these negative images are matched to each logo template and a transform parameter instance $(\Delta\sigma, \Delta\theta)$ is recorded for each pair. Finally, a joint probability density of scale and orientation change is estimated to represent the matching statistics of the background. This density is called the Background Model, and will be further elaborated in Section 4.3.2.2.

Detection stage (Figure 4.23b), similar to the training stage, consists of a preprocess applied on each test image followed by a codeword-based appearance match. These matches contribute to the transform parameter density estimation of the test image. In this density, regions



(a)



(b)

Figure 4.23: Algorithm Flow Diagram (a) Training Phase, (b) Detection Phase. $T1$, $T2$ and $T3$ stages of the training phase are explained in Sections 4.3.2.1, 4.3.2.1 and 4.3.2.2. $D1$, $D2$,$D3$,$D4$ and $D5$ stages of the detection phase are explained in Sections 4.3.2.1, 4.3.2.1, 4.3.2.3, 4.3.2.3 and 4.3.2.3.

that are highly unlikely according to the background density are identified. These regions then undergo detailed analysis based on a combined representation called *Combined Visual Knowledge Base*, which has already been defined in Section 4.2.

4.3.2.1 Local Feature Detection, Description and Appearance Quantization

Preliminary stages of the extended method introduced in this section, utilizes the same algorithms for local feature detection and description as the initial version, DoG detector (Section 2.1) and SIFT descriptor [75]. Appearance of local features are also quantized using the same codebooks used in Section 4.2.

4.3.2.2 Background Model Estimation

The aim of this step is to model the distribution of transform parameters, scale and orientation change among images that contain no common properties. In order to achieve this, template image is compared to a database of background images. The comparison involves the following steps:

1. Accept each background keypoint to each template keypoint, if they have the same codewords.

2. Compute the difference between scale and orientation of these matching pairs.

3. Estimate the joint probability density of scale and orientation changes in each of the background images.

4. Average and normalize these densities to obtain the background model, $p_{bg}(\sigma, \theta)$.

The function of background density estimation from negative training images is to model bias in matching process of ideally scale invariant and orientation covariant local descriptors. In the ideal case, comparison of logo templates with random images is expected to result in a uniform 2D density of scale and orientation changes. However, in real case, this background density is biased (Figure 4.24a), since interest points tend to match to correspondences with similar scales.

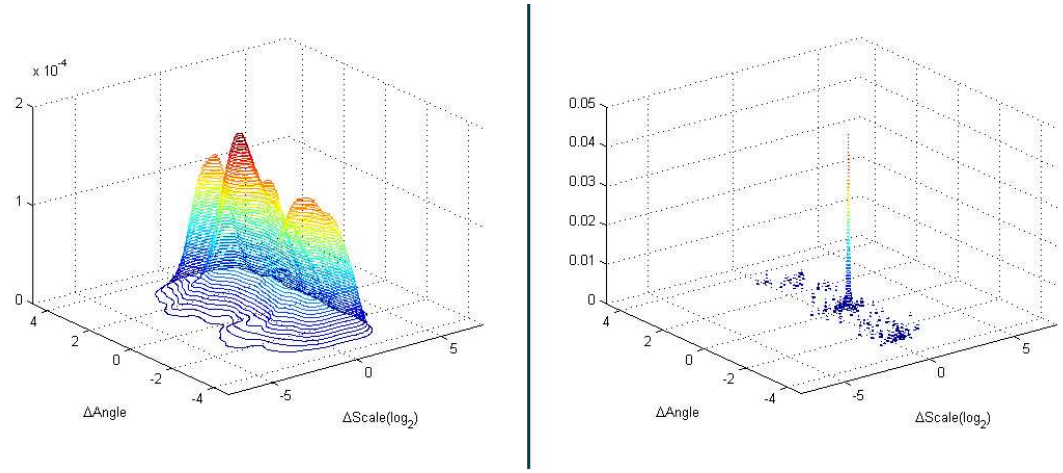


Figure 4.24: Example joint transform densities of random (Left) and true match (Right) cases.

4.3.2.3 Matching Groups of Keypoints

Groups of keypoints between two images one being the template is achieved in two steps. In the first step, matches are obtained by comparing merely quantized codewords of keypoints. Most of the matches at this point is erroneous, and need to be carefully filtered before delving into computationally complex steps. This is performed by the *a contrario* technique [175]. The second step processes each of the meaningful groups of keypoints using affine-invariant methods and increases the level of confidence.

Transform Density Estimation and A Contrario Group Identification In this step, similar to Section 4.3.2.2, joint probability density of scale and orientation difference is estimated between a test image and a template image whose background model has already been obtained. This density, $p_{test}(\sigma, \theta)$, estimated from the scale and orientation change ($\Delta\sigma$ and $\Delta\theta$) coarsely shows the grouping of these transform parameters. Although not very frequent, in ideal cases looking to this density is enough to discriminate a true match from a false one (Figure 4.24). However, most of the time, transform density is somewhere in between these two extremes that are given in Figure 4.24. It is worth mentioning that this distribution, in fact, is used to identify significant groups of keypoints that undergo similar scale and orientation changes. In addition, local features that are on the background of the object under consideration will also contribute to this distribution. Therefore, in cases, where a substantial amount of background is overlapping between images, peaks of the transform density will be

dominated by the background. This fact is alleviated in our method by modeling the query object using uncluttered ideal images.

In a non-ideal world, we have to identify the highly meaningful clusters (peaks) of transform density between two images. In order to achieve this, first we contrast these densities:

$$p_{diff}(\sigma, \theta) = p_{test}(\sigma, \theta) - p_{bg}(\sigma, \theta) \tag{4.4}$$

where p_{bg} is the background transform density estimated as in Section 4.3.2.2.

Prominent peaks of p_{diff} shows the positions (i.e. parameter pairs) around which test image matches exhibit clustering behavior that is highly unlikely for background matches. These positions are interpreted as possible centers of highly meaningful clusters. Around each of these cluster centers, for regions with various sizes, Number of Expected False Alarms (NFA) [175] is computed using the background density and the number of test image matches, whose transform parameters are in the corresponding region:

$$NFA\,(X, R) = N_R \cdot M \cdot B\left(M, K(X, R), P_{bg}(R)\right) \tag{4.5}$$

X is the cluster center coordinate $(\Delta\sigma_c, \Delta\theta_c)$, R is the region around center, $K(X, R)$ is the number of test image match transform parameter instances $(\Delta\sigma, \Delta\theta)$ that fall into region R, M is the number of all $(\Delta\sigma, \Delta\theta)$ instances, and $B(\cdot)$ is the tail of the binomial law defined by:

$$B(M, k, P) = \sum_{j \geq k} \left(\begin{array}{c} M \\ j \end{array} \right) P^j \,(1 - P)^{M-j} \tag{4.6}$$

The highest of the NFA values computed for regions around a cluster center is assigned to it and the corresponding radius. All clusters are then evaluated according to their NFA values, and clusters that have a value over the threshold are filtered out. This way, a contrario elimination of insignificant clusters randomly is achieved.

Combined Appearance and Affine Geometry Description of Keypoints Clusters that have significantly low NFA value are accepted as candidates for consistent groups of matching

keypoints. These groups, however, are still highly ambiguous and should be further evaluated at a lower level. In order to achieve this aim, a combined representation involving affine invariant geometrical descriptors and quantized appearance descriptors is utilized. The details of this representation has already been presented in Sections 4.2.2.3 and 4.2.2.4.

Decision based on Combined Visual Knowledge Base A test keypoint is accepted as a match to a template keypoint, if its CVKB consisting of highly discriminative quads show resemblance to its template counterpart. The comparison of CVKB is explained in this section.

The structure of each of these representations can be clearly explained as follows: Let CVKB consist of N quads, q_i, $i \in 1, \cdots, N$. Each quad has two elements, $\vec{cw}_i$ and $\vec{b_i}$. $\vec{cw}_i$ is the vector of codewords cw_{ij} corresponding to interest points q_{ij}, where $j \in 1, \cdots, 4$. Barycentric coordinates $\vec{b_i}$ of each quad is a 3-dimensional vector composed of b_{ik}, where $k \in 1, \cdots, 3$. This leads to the following representation for a quad:

$$q_i : \left(\vec{cw}_i, \vec{b_i}\right) \tag{4.7}$$

Quads within test keypoint CVKB are compared to the template CVKB by this representation. The comparison is performed for each field of the representation separately. Let q_t be the test quad to be compared to CVKB. The first thing to be compared is the codewords of the two quads, if they do not match, then there is no need to compare other fields. If the codewords match then barycentric coordinates are compared. Let q_i be defined as in Equation (4.7) and q_t be defined as $(\vec{cw}_t, \vec{b_t})$. Compatibility of these two quads is assessed as follows:

1. If $\forall j \in 1, \cdots, 4, cw_{ij} = cw_{tj}$, then continue; else quads are incompatible.

2. Compute barycentric coordinate distance d_b by using Equation (3.28).

3. q_i is compatible with q_t, if and only if $d_b < thr_b$

Two CVKB are accepted as compatible, if a predefined number of their quads (i.e. thr_{quad}) are compatible. Two keypoints need to have compatible CVKB's in order to be accepted as matches.

Table 4.6: Simulation parameters for the proposed and the baseline (Algorithm 4.1) methods

	Parameters	Min. Value	Max. Value	Step Size
Proposed	thr_{NFA}	-1	15	1
	thr_{bary}	0.01	0.5	0.01
	thr_{quad}	0	50	1
Baseline	Distance Ratio	0.6	0.8	0.1
	# Pt. Matches	0	250	1

4.3.3 Evaluation: Joint Utilization of Appearance, Geometry and Significance-based Grouping for Scene Logo Retrieval

In order to perform the preliminary evaluation of the proposed algorithm the dataset that is used for evaluation of the previous method is utilized. The visual codebook created for assigning appearance descriptors of test and template images consists of 128 codewords, and it is the same as the one used in Section 4.2.

Performance values are measured in terms of true positive and false positive rates. The result of these measurements are presented via Receiver-Operating Characteristic(ROC) curves. A test image is identified as a match, if it has a meaningful transformation cluster which has an NFA value lower than thr_{NFA}, and at least one keypoint with a matching CVKB according to the two parameters thr_{bary} and thr_{quad}. The range of the parameter search space is given in Table 4.6.

In addition to the simulation case using two different template logos (Figure 4.25), a second simulation for observing the performance of the proposed method with an excessive number of templates (Figure 4.26) is performed. This extension, which is denoted as Artificial Template Extension (ATE) throughout the text, consists of artificially applying a number of viewpoint changes and scale transforms (six and four, respectively) on template images. These newly created images are then used in the same manner as the original template images.

In both simulations performance of the proposed algorithm is also compared with a baseline method (Algorithm 4.1), that is prominent in the literature. The baseline algorithm is the most recommended matching method for SIFT [75]. This algorithm also has a geometric compatibility checking stage to compansate for affine changes, and due to its nature directly

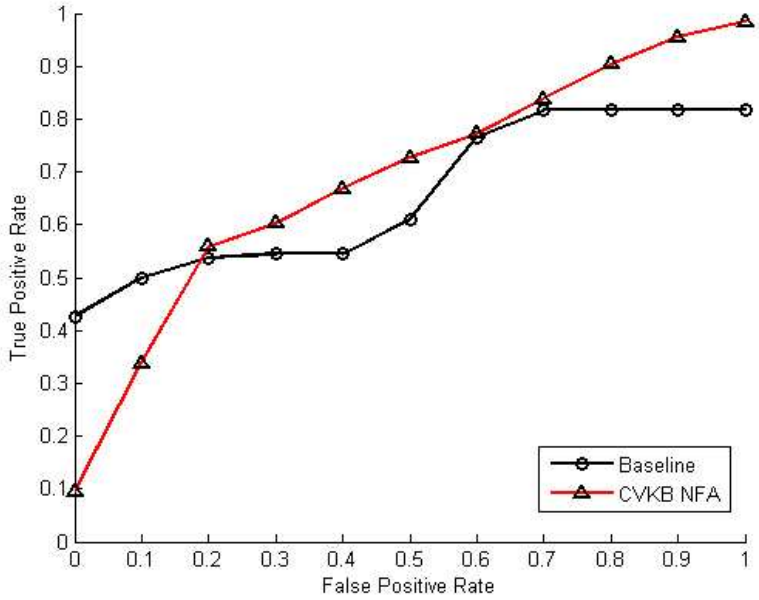


Figure 4.25: Performance results of preliminary simulations using original templates (without ATE)

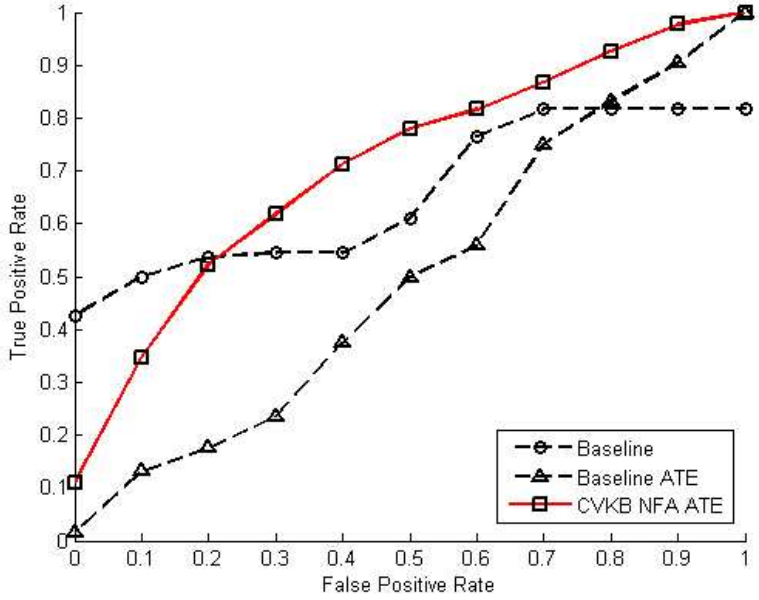


Figure 4.26: Performance results of preliminary simulations using ATE with the proposed algorithm and comparison to baseline (Algorithm 4.1)

Figure 4.27: Sample images from the Belga Logos Dataset [152].

compares SIFT descriptors without quantization (Section 2.2). There are two parameters that are varied to analyze the performance of the baseline algorithm. The first of these parameters is the ratio of the first nearest matches descriptor distance to the second. Any match having a ratio higher than the threshold is rejected. Three threshold values, $0.6, 0.7$ and 0.8 are used during the experiments. The other parameter used to evaluate the performance of the baseline algorithm is a lower limit on the number of matches that survived at the end of the geometric compatibility process. A much wider range of $[0, 250]$ is used for this parameter.

Experiments performed also include utilization of the stop-list approach [168] that is frequently applied in text retrieval. In a seperate strand of experiments, very frequent codewords and quads are detected and dismissed before applying the proposed algorithm. Despite its success in text retrieval domain, this approach does not enhance the performance of the proposed algorithm in this domain based on the conducted experiments. Therefore, the experimental results related to these experiments are not included in this thesis.

In the light of the preliminary evaluation of the algorithm and the ATE extension, a second set of experiments is conducted. Belga Logos [152], a recently introduced dataset, is used for testing the algorithm on logos of three more brands, namely “Kia”, “Citroen” and “Base”, in addition to the previous “Coca-Cola”. For a controlled evaluation, instead of all the database, a subset of size 500 is created (Figure 4.27). Experimental data consists of non-overlapping positive images (only one of the searched logos is present) of the four brands, and negative images having context similar to these images. The number of positive images for “Coca-Cola”, “Kia”, “Citroen” and “Base” logos are 32, 82, 46 and 136, respectively.

In this second stage of experiments, two methods that have excelled in the previous experiments, baseline with single template and proposed method with ATE are compared. ATE extended version of the baseline method, and the single template version of the proposed method is not included in the performance visualizations for simplicity, since they performed

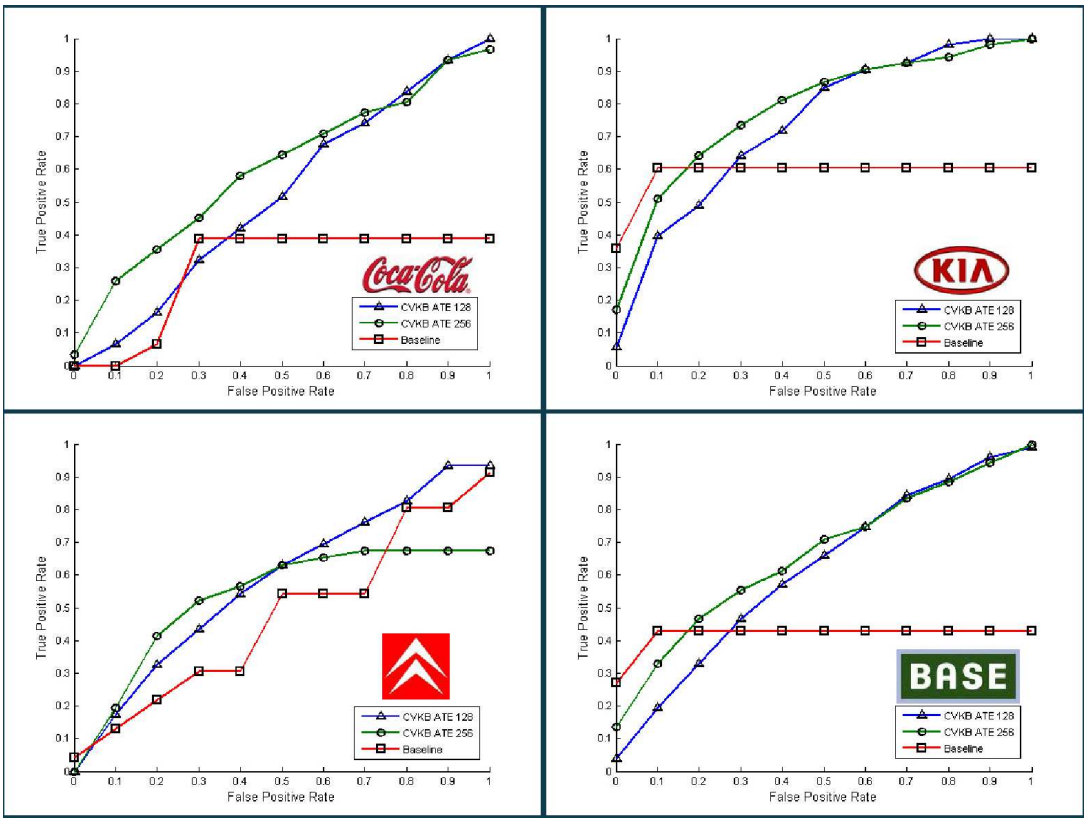


Figure 4.28: Performances of the CVKB ATE (with 128 and 256 codebook sizes) and baseline algorithm (non-ATE) on four logos of the Belga Logos Dataset. (Upper Left) Coca-Cola, (Upper Right) Kia, (Lower Left) Citroen, (Lower Right) Base.

similar to the previous experiments and provided no additional information. Instead, in order to evaluate the effect of increasing the codebook size, a second version of the proposed algorithm working on a codebook of size 256 is included.

The number of artificial templates created for each logo in this experiment is set to six (original and five different viewpoints), since template logos provided by the Belga Logos have relatively small sizes. This decision is based on the detailed analysis of previous experiments, where it is observed that templates below a given size do not provide any useful information. The results of this second set of experiments are given in Figure 4.28 as separate RoC curves for each logo.

4.3.4 Conclusions

A template-based matching approach for matching distinctive groups of interest points has been presented. Distinctiveness is modeled in a probabilistic way in terms of transformation

parameters, and compatible groups of true keypoint matches are identified. This is achieved by utilization of the expected number of false alarms metric for a given group of interest points. This approach adds a low level matching stage to *a contrario* decision method and enhances the robustness of appearance representation by using clustered local descriptors, while compensating the side effect of decreased individual discriminative power by the help of group-based geometrical constraints.

The *a contrario* decision method, which is previously applied to level line shape descriptors, gave promising results in that context. Applying it to the keypoint domain is also straightforward. In this context, however, we have applied this method in the domain of keypoints whose appearance is represented by generic quantized codewords. This is an alternative to the classical methods which should handle the burden to compare every query keypoint appearance descriptor with the whole database of descriptors.

In the first stage of experiments, two different scenarios are demonstrated. The first one presents the performance of the approach using only the two original template images. Except for a small part, proposed approach is superior than the widely accepted and adhered baseline method (Algorithm 4.1). It should be noted that the baseline algorithm compares the keypoint appearances using far more detail, while our approach uses only 128 codewords. Due to the harsh transformations in the deliberately selected challenging positive set, the performance of the algorithms using only perfect templates is limited. An obvious improvement is to extend the search using multiple versions of the sought image. Although this approach puts a great burden on the baseline algorithm and its derivatives in terms of computational power, the resulting improvement in true positive rate is mostly overshadowed by the skyrocketing false positive rate. Artificially transformed templates included in the search, in addition to the original does not affect the internals of the search mechanism. Each of these extended query images is searched in the database independently, only affecting the final decision on the test image. Therefore, the uncontrollable surge in the false positive rate indicates the lack of scalability, i.e., ability to increase recall performance via increasing training (template) data. This is an obvious consequence of increased chance of match to an increased size of query set and common for methods relying on nearest descriptor search.

On the other hand, in the proposed algorithm, the increase in true positive rate with the introduction of new query samples is not accompanied by an unacceptable increase in false

positive rate. Scalability of the compared algorithms is best observed at the second stage of experiments that are conducted on the Belga Logos Dataset. In this dataset, context (clutter, background texture, reflectance, etc.), quality (illumination, focus, etc.) and viewpoint have a much higher variance than the dataset used for the first stage experiments. Under these conditions, matching a keypoint on a logo instance (for example on a textured non-rigid T-shirt viewed with an angle), to an ideal logo template with a descriptor distance significantly lower than any other keypoint, may not be possible. This is due to the limits of invariance/covariance of local descriptors under changes in viewpoint, illumination, etc. In order to ameliorate the recall performance in these situations, one should add more training instances or queries representing as much variance as possible. This phenomenon can be best observed in the performance results of Coca-Cola logo (Figure 4.28). Since baseline algorithm can not benefit from ATE, similar to the proposed algorithm, its recall has a much lower limit.

Another important problem with the methods, which rely on an initial set of precise appearance matches for detecting geometrically significant group of keypoints, can also be observed in the second dataset. During the analysis of the Coca-Cola and Citroen logo detection results, many examples of "miss" have been observed in images where multiple instances of these logos exist. This is a direct result of primitive nearest neighbor-based descriptor matching algorithm. Nearest distance descriptor matching aims at reaching precise one-to-one matches between template and test images before considering any collective geometrical property. This in turn leads to many match groups with inadequate populations, sparsely distributed on different instances of the same logo, in test images containing multiple instances of the query logo.

During the experiments, it is observed that miss-detections arise mostly in images, where searched logo undergoes significant affine transformations. For those cases, detailed analysis shows that the initial quantized appearance matching fails due to extreme deformations in the local appearance pattern. This fact degrades the performance of the algorithm significantly, since the method is based on the assumption that initial correspondences exist between model and query images. Therefore, the appearance matching component of the proposed method is open to improvement in increasing its robustness against geometric transformations. In order to achieve this, a promising alternative is the utilization of *affine adaptation* methods [159, 176] that allow for more accurate descriptor extraction via incorporating a preliminary local patch adaptation step.

The proposed method, in the light of the experiments, provides a robust way to achieve template matching with large template sets and using much less discriminative yet repeatable local features. The approach has some obvious points of improvement such as the usage of multiple templates. Currently, every template is searched on its own, although these can easily be linked together, enabling keypoints from multiple template images be matched as a group [177]. Another point is the opportunity to utilize background modeling approach at the quad matching point to objectively decide barycentric distance threshold for each template separately. Improvements in the lower levels may include empirical estimation of the optimal codebook size using a systematical procedure and a reasonable search space, which is an issue deliberately omitted in this thesis. Last, but not the least, the performance of the method can profit significantly from the improvement of the appearance matching step with affine adaptation methods. In the light of experiments, the proposed approach, despite having various points that require improvement, provides an efficient method of local logo-like template search without the need for computationally intensive detailed appearance comparison. This property is crucial for todays large databases such as the one used in TRECVID [167], which are most of the time indexed using codeword based descriptions.

CHAPTER 5

JOINT UTILIZATION OF APPEARANCE AND GEOMETRY FOR 3D OBJECT RECOGNITION

In this chapter the problem of 3D object recognition from visual content is considered. As a possible solution to this problem, we propose a novel method for 3D object recognition, which utilizes well-known local features in an efficient way, without any reliance on partial or global planarity. Geometrically consistent local features, which form the crucial basis for object recognition, are initially identified by using affine 3D geometric invariants [149]. The utilization of 3D geometric invariants replaces the classical 2D affine transform estimation/verification step [176, 178, 75, 179, 49], and provides the ability to directly verify 3D geometric consistency. The main contribution of the proposed approach lies in this ability of incorporating highly discriminative affine invariant 3D information much earlier in the process of matching in comparison with its counterparts. The accuracy and robustness of the method in highly cluttered scenes, without any prior segmentation or post 3D reconstruction requirements, are presented in the experiments.

5.1 Motivation

The method presented in this chapter addresses the problem of general object recognition from 2D images in a novel approach. In doing so, it differentiates itself from the traditional methods in the literature with its approach to geometric verification of low-level matches that are based on local appearance descriptions Table 5.1. The proposed method exploits 3D and 2D geometric invariants, which has its roots in the previous decade, in a simple setting without resorting to complex computations of 3D model estimation.

Table 5.1: Main Components of Some Representative 3D Object Recognition Methods

	Appearance Matching	**Geometric Verification**	**Applicability**
Rothganger et. al. [176]	DoG, Harris-Laplace Affine Adaptation NCC, SIFT, Color Potential Match List Neighboring Matches	Local Patch Geometry Projection Matrix Est. Chaining Bundle Adjustment Euclidean Upgrade	Partially planar objects viewed under weak perspective conditions
Weiss et. al. [149]	None	3D & 2D Invariants	3D Objects with distinctive intersecting edges
Hough-based (Alg. 4.1) [75]	DoG & SIFT (Section 2.1.2 & 2.2) Best Match Only	Hough Trans. Clust. 2D Aff. Trans. Est. (Algorithm 4.1)	Planar Objects Single instance only Limited appearance distortion
Proposed Method	DoG, Harris-Affine SIFT	3D & 2D Affine Invariants	Generic 3D objects viewed under weak perspective conditions

Recognition of objects has been the focus of computer vision systems for almost half a century. However, despite decades of intensive research, real life object recognition systems still has many constraints for successful recognition [1]. At the heart of this fact, lies the difficulty of adapting the optimization boundary between discriminative power and invariance against a broad spectrum of real life attacks that effect the appearance of objects [180]. Today, utilization of local features and local appearance based methods, is the widespread approach to this problem [176, 178, 75, 179, 49]. Local features dominate other approaches with some properties that render them well suited to complex recognition applications, such as object recognition. Their significance is twofold: First, they provide a robust way to represent the images in terms of parts without the need for segmentation. Second, they provide a computationally feasible number of well localized and individually identifiable anchor points. Local features depend on the common assumption that sufficiently small patches on 3D scenes can be treated as being comprised of planar points [176]. This assumption is valid for patches of small sizes and local features that are detected by current state-of-the-art detectors mostly comply with this size restriction. Local features are, therefore, directly applicable to planar object recognition without extra restrictions [44, 45].

Recognizing 3D objects from 2D images, however, is a more difficult task when compared to planar object recognition in 2D images. In addition to the obvious self occlusion problem, 3D objects undergo a more complex transformation during the imaging process. Due to the loss of 3D information during this complex process, which is broadly called 3D to 2D projection, general 3D objects has no invariants that may enable direct recognition of feature groups from their 2D images [36]. This fact is widely accepted in modern methods and, therefore, geometric verification/recognition is conveniently handled by imposing some restrictions on the viewing conditions. Under particular cases, where objects are viewed from a distance much larger than the relief of the object, planarity assumption is extended to groups of small patches. In methods embracing this idea [176, 178, 75], special groups, which are compatible with the planarity assumption, are detected between model and test images using a 2D affine transform estimation/verification step. This step is followed by addition of extra matches that are compatible with the estimated transform [176]. The performance of these methods, which are dependent on these restrictions, are inherently limited at the initial feature grouping process in unconstrained cases where 3D characteristics of the object can not be neglected. The shortcomings of the grouping process, are typically solved in later steps of recognition, which involve utilization of 3D reconstruction methods, such as *pose estimation* and addition of new matches based on *reprojection* [176]. In more recent approaches [181], 3D information is directly incorporated into the feature extraction process via using *depth maps* as an extra information source.

Despite the difficulties that today's systems experience in the process of 3D object recognition in the absence of an extra modality, such as depth maps, the inspiring human visual system experiences minimum, if any, problem in recognizing 3D objects from their 2D photographs. This success had once been the driving force behind the efforts in the strand of geometric object recognition and eventually resulted in creation of a strong mathematical background on the subject geometric invariants [1]. Despite the discouraging fact that there are no geometric invariants of 3D to 2D projection [36], the resulting research, which is based on previous decade's strong but deserted geometric invariance field, has proved fruitful in finding alternative approaches. A novel example of these methods [149] utilizes invariant information in matching 3D object models to their projections in images.

Our research, aims to bridge the gap between geometric information conveyed through useful relations among 3D-3D and 2D-2D geometric invariants (Section 5.2) and robust appear-

ance information conveyed through covariant local feature descriptions. In doing so, the proposed method builds on the foundations constructed by the work of Weiss and Ray [149] and presents a novel method for 3D object recognition, which utilizes well-known local features in a less restricted, and therefore more efficient, way. In the presented approach, geometrically consistent local feature groups, which form the crucial basis for object recognition, are identified using affine 3D geometric invariants. Utilization of 3D geometric invariants replaces the aforementioned traditional 2D and then 3D affine transform estimation/verification step (Table 5.1), and provides the ability to directly verify 3D geometric consistency. The main contribution of the proposed approach lies in this ability of incorporating highly discriminative affine 3D information much earlier in the process of matching in comparison with its counterparts. This way, complex 3D model extraction stages can be replaced with a much simpler geometric consistency step. The accuracy and robustness of the method in highly cluttered scenes, without any prior segmentation or post 3D reconstruction requirements, are presented in the experiments.

This research aims to bridge the gap by deriving invariant relations between geometric invariants of 3D-3D (Section 3.3.2) and 2D-2D (Section 3.3.1) invariants [149]. These invariant relations that exist between invariants are developed depending on specific projection assumptions in Section 5.2. Next, in Section 5.3, preliminary simulations for assessing the performance of these invariant relations on controlled datasets are presented [50]. The proposed novel 3D object modeling and recognition framework, which builds upon the conclusions drawn from the preliminary simulations, is described in Section 5.4. Section 5.4.3 following the method description demonstrates utilization of the proposed method for recognition in cluttered scenes making use of experimental results obtained on a prominent dataset [50, 51]. In the last section (Section 5.5), performance of the proposed method is reviewed in the context of the experimental results and significant details that may shed light on future research are discussed.

5.2 Invariant Geometric Relations of 3D to 2D Projection

In this section, the results of a previous research [149] on deriving invariant relations between 3D and 2D invariants are reproduced. These results include relations constructed under two different camera projection approximations, namely, perspective and weak-perspective.

Perspective and weak-perspective camera approximations are shown to be composed of projective and affine components respectively in Section 3.2. Therefore, the perspective camera assumption and projective case are used interchangeably in the following text. The same duality is used when referring to the weak-perspective camera model and affine case.

5.2.1 Invariant Relations of 3D and 2D Invariants under Perspective Camera Projection

Invariants of 3D to 3D and 2D to 2D projective transformation have been derived in Section 3.3. In this section, the constraining relations between invariants of 3D and 2D projective transformations are derived, briefly reproducing the work of Weiss and Ray [149] (see also Appendix A). Although 2D projective invariants exist for point sets of size 5 (Section 3.3.1.2), due to the requirement of having at least 6 points for 3D invariants, these relations are derived for 6 pairs of points in homogeneous coordinates.

In Section 3.3.2, points $\vec{X_5}$ and $\vec{X_6}$ have been expressed as a weighted sum of the first 4 points $\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}$:

$$\lambda_5 \vec{X_5} = a\lambda_1 \vec{X_1} + b\lambda_2 \vec{X_2} + c\lambda_3 \vec{X_3} + d\lambda_4 \vec{X_4} \tag{5.1}$$

$$\lambda_6 \vec{X_6} = a'\lambda_1 \vec{X_1} + b'\lambda_2 \vec{X_2} + c'\lambda_3 \vec{X_3} + d'\lambda_4 \vec{X_4} \tag{5.2}$$

The unknowns λ_i in the above equations, which appear in the form $a\frac{\lambda_1}{\lambda_5}, \cdots d\frac{\lambda_4}{\lambda_5}$ and $a'\frac{\lambda_1}{\lambda_6}, \cdots d'\frac{\lambda_4}{\lambda_6}$ in two sets of four equations organized for defining $\vec{X_5}$ and $\vec{X_6}$ can be eliminated using the cross ratios of determinants. In order to achieve this aim, let us denote determinant of the 4×4 matrix formed by an ordered quadruple of 3D point coordinates, as M_i. For instance, determinant of the matrix formed by the first 4 points, $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4})$, can be defined as:

$$M_5 = \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| \tag{5.3}$$

Using this convention, the 3D projective invariants I_1, I_2 and I_3 of points $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5}, \vec{X_6})$ are defined by using coordinates of points $\vec{X_5}$ and $\vec{X_6}$, i.e. (a, b, c, d) and (a', b', c', d') in Equations (5.1) and (5.2) as:

$$I_1 = \frac{ab'}{a'b} = \frac{M_1 M_2'}{M_1' M_2}, \quad I_2 = \frac{ac'}{a'c} = \frac{M_1 M_3'}{M_1' M_3}, \quad I_3 = \frac{ad'}{a'd} = \frac{M_1 M_4'}{M_1' M_4} \tag{5.4}$$

M_i' terms in the above equation represent a determinant of the 4×4 matrix formed by an ordered quadruple of 3D point coordinates $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_6})$, instead of $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5})$.

Now, we need to derive 2D-2D projective invariants of 6 points $\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_5}$ and $\vec{x_6}$ that are homogeneous representations of projections of 3D points $\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5}$ and $\vec{X_6}$. Following the convention defined in [149], square matrices that are of size 3×3 can be defined using a triplet of ordered point coordinates in 2D space selected from a group of 5 points. The determinant of this matrix, is then named using the indexes of the remaining two points. For example, the determinant of the matrix formed using 2D coordinates of points $\vec{x_3}$, $\vec{x_4}$ and $\vec{x_5}$, selected from the 5 point group $(\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_5})$ is defined as:

$$m_{12} = \left|\vec{x_3}, \vec{x_4}, \vec{x_5}\right| \tag{5.5}$$

Using this convention, it is proved in Appendix A.3 that four 2D projective invariants of the a point set $(\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_5}, \vec{x_6})$ are defined as:

$$i_1 = \frac{m'_{12} m_{14}}{m_{12} m'_{14}}, \quad i_2 = \frac{m'_{12} m_{35}}{m_{25} m'_{13}}, \quad i_3 = \frac{m'_{12} m_{13}}{m_{12} m'_{13}}, \quad i_4 = \frac{m'_{12} m_{45}}{m_{25} m'_{14}} \tag{5.6}$$

In the above equations, m'_{ij} is defined exactly the same as m_{ij}, except for replacing point $\vec{x_5}$ in the 5 point set by $\vec{x_6}$ and define the point set as $(\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_6})$. The relation between 3D and 2D projective invariants, namely (I_1, I_2, I_3) and (i_1, i_2, i_3, i_4) can be obtained through a series of substitutions [149] as:

$$I_3(I_2 - 1)i_1 i_2 - I_3(I_1 - 1)i_1 - I_1(I_2 - 1)i_2 = I_2(I_3 - 1)i_3 i_4 - I_2(I_1 - 1)i_3 - I_1(I_3 - 1)i_4 \tag{5.7}$$

Details of the derivations in this section are given in Appendix A.3.

5.2.2 Invariant Relations of 3D and 2D Invariants under Weak-Perspective Camera Model

The invariants of affine transformation in 3D space (12 degrees of freedom) requires at least 5 non-degenerate points for computation (Section 3.3.2). Computing 2D affine invariants, however, is possible with only 4 points, since the corresponding transformation has only 6 degrees of freedom (Section 3.3.1). In order to construct the relations between 3D and 2D invariants, we should meet the minimum requirements for calculating both sets of invariants. For this reason, the formulated relations involve 5 3D scene point coordinates $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5})$ and respective coordinates of their 2D projections $(\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_5})$. In Section 3.3.2, point $\vec{X_5}$ have been expressed as a weighted sum of the first 4 points $\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}$:

$$\vec{X_5} = a\vec{X_1} + b\vec{X_2} + c\vec{X_3} + d\vec{X_4} \tag{5.8}$$

Coefficients (a, b, c, d) in this representation are invariants, which satisfy the equation $a + b + c + d = 1$ and stay unchanged under 3D affine transformations. As shown in [149], three of these invariants are independent, and they can be computed using determinants, which are also relative invariants of projective transformations. The determinant of the 4×4 matrix formed by an ordered quadruple of 3D point coordinates is denoted as M_i, similar to the projective case (Section 5.2.1). Definition of M_5 is given in Equation (5.3) as an example. Using this convention, 3D affine invariants can be expressed in terms of determinants as given in Section 3.3.2:

$$I_1 = \frac{M_1}{M_5}, \quad I_2 = \frac{M_2}{M_5}, \quad I_3 = \frac{M_3}{M_5}$$

Following the convention in Equation (5.5) in Section 5.2.1, square matrices that are of size 3×3 can be defined using a triplet of ordered point coordinates in 2D space. Using these definitions, 2D affine geometric invariants, i_1, i_2, i_3 and i_4 of points $(\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_5})$ can be represented as ratios of determinants as [149]:

$$i_1 = \frac{m_{12}}{m_{15}}, \quad i_2 = \frac{m_{13}}{m_{15}}, \quad i_3 = \frac{m_{25}}{m_{15}}, \quad i_4 = \frac{m_{35}}{m_{15}} \tag{5.9}$$

The relation given in Equation (5.8) between 3D scene points, also holds among their 2D projected coordinates. In order for this to be true, the only requirement is that 3D to 2D projection should take place in conditions where weak perspective (affine) camera model assumption holds. Under these constraints, 5 2D image points, $(\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_5})$ which are projections of scene points $(\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5})$, are related through Equation (5.10):

$$\vec{x_5} = a\vec{x_1} + b\vec{x_2} + c\vec{x_3} + d\vec{x_4} \tag{5.10}$$

The 3D and 2D invariants obtained from the 3D and 2D homogeneous point coordinates, respectively, are connected to each other through Equations (5.8) and (5.10). Under weak-perspective projection assumption, the coefficients (a, b, c, d) remain the same for both 3D and 2D projected coordinates of world points. Using this fact, and performing a series of substitutions and simplifications, one can obtain the relation between 3D and 2D affine invariants:

$$i_1 + I_1 i_3 - I_2 = 0 \tag{5.11}$$

$$i_2 + I_1 i_4 - I_3 = 0 \tag{5.12}$$

The derivation and proof of these relations are given in Appendix A.2. Utilization of these two equations for accomplishing the task of matching images of 3D objects, on the other hand, is explained in the following sections.

The implications of these equations are better understood by analyzing their geometrical meaning in the domain of 3D geometric invariants. Equations (5.11) and (5.12), each by itself represent a plane in 3D invariant space (I_1, I_2, I_3). The geometric entity that satisfy these two plane equations in a typical degenerate case is a line, which defines their intersection. In other words, each 5 point combination in an affine projection of a 3D scene to 2D imposes constraints on possible values of 3D invariants of the corresponding 3D scene points. These constraints limit the location of 3D invariants on a line, whose parameters are defined by 2D invariants (i_1, i_2, i_3, i_4) of the image points. Each projection of the same scene imposes the same type of constraints on 3D invariants of the real scene points. This also means that, 3D invariants of the scene must satisfy all of the line equations simultaneously, and therefore, reside at the intersection point of all these lines. This fact will be used in experiments in Section 5.3.

Table 5.2: 3D coordinates of 6 points that are used in the simulations of Section 5.3.1

	3D World Point Coordinates					
	$\vec{X_1}$	$\vec{X_2}$	$\vec{X_3}$	$\vec{X_4}$	$\vec{X_5}$	$\vec{X_6}$
x	0	4	0	0	2.8	-2
y	0	0	4	0	-0.4	2.8
z	4	0	0	0	2	1.6

Table 5.3: 2D projected coordinates of 6 points using the middle view

	2D Coordinates of Projected 3D World Points					
	$\vec{x_1}$	$\vec{x_2}$	$\vec{x_3}$	$\vec{x_4}$	$\vec{x_5}$	$\vec{x_6}$
x	0	-14.5042	26.1784	0	-13.4914	23.7127
y	24.1737	-19.339	-8.7261	1.7e-014	-0.4997	10.8163

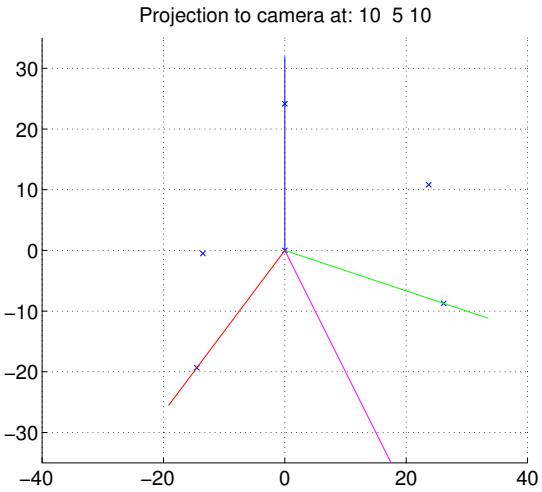
5.3 Preliminary Experiments

5.3.1 Robustness Analysis of Projective Invariants

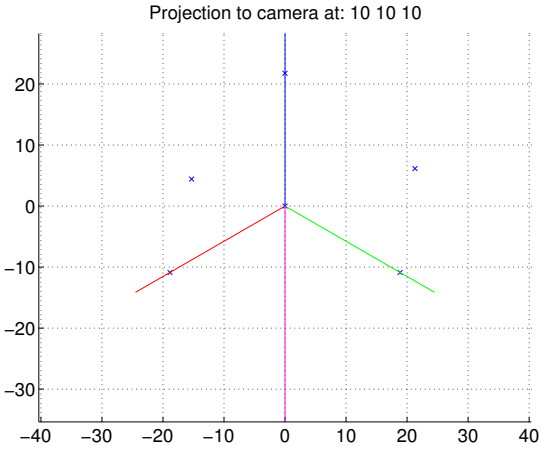
In the first set of simulations, projective invariants that are resistant against perspective transformation were tested. In Figure 5.1, projection of 6 3D points, first 4 of which is not coplanar are given. Projections are obtained from three different point of views. 3D Invariants are calculated from coordinates of artificially located points. On the other hand, three sets of 2D invariants were obtained from three point of views with a perspective camera. Numerical values for homogeneous 3D point coordinates are given in Table 5.2, while 2D coordinates of the projection obtained from the second camera are given in Table 5.3.

Projective 3D to 2D relation (Equation 5.7) is tested for robustness against pixel errors and the results are given in Figure 5.2 and Figure 5.3. These figures present the plots of the absolute error in the equation when one of the non-base points is perturbed in terms of percentage and absolute pixel coordinates, respectively.

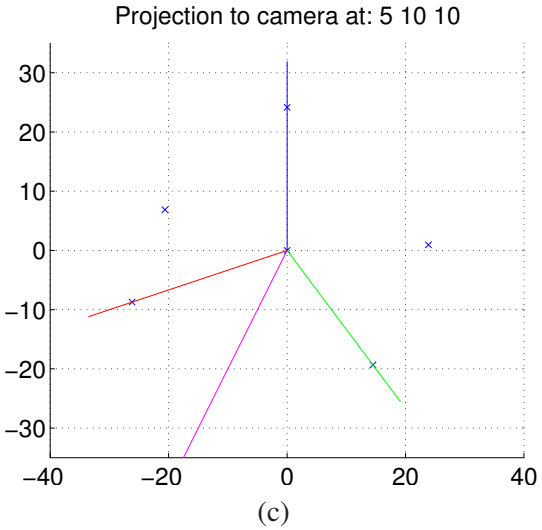
It can be deduced from the error curves that projective invariants are quite fragile against perturbations in 2D projection. This is definitely an important weakness, that should prohibit its use in imperfect conditions, such as real life scenarios.



(a)



(b)



(c)

Figure 5.1: Projection of six 3D points from three different viewpoints. All cameras look towards origin from the coordinates indicated above the figures. (*a*) Camera at (10,5,10), (*b*) Camera at (10,10,10), (*c*) Camera at (5,10,10)

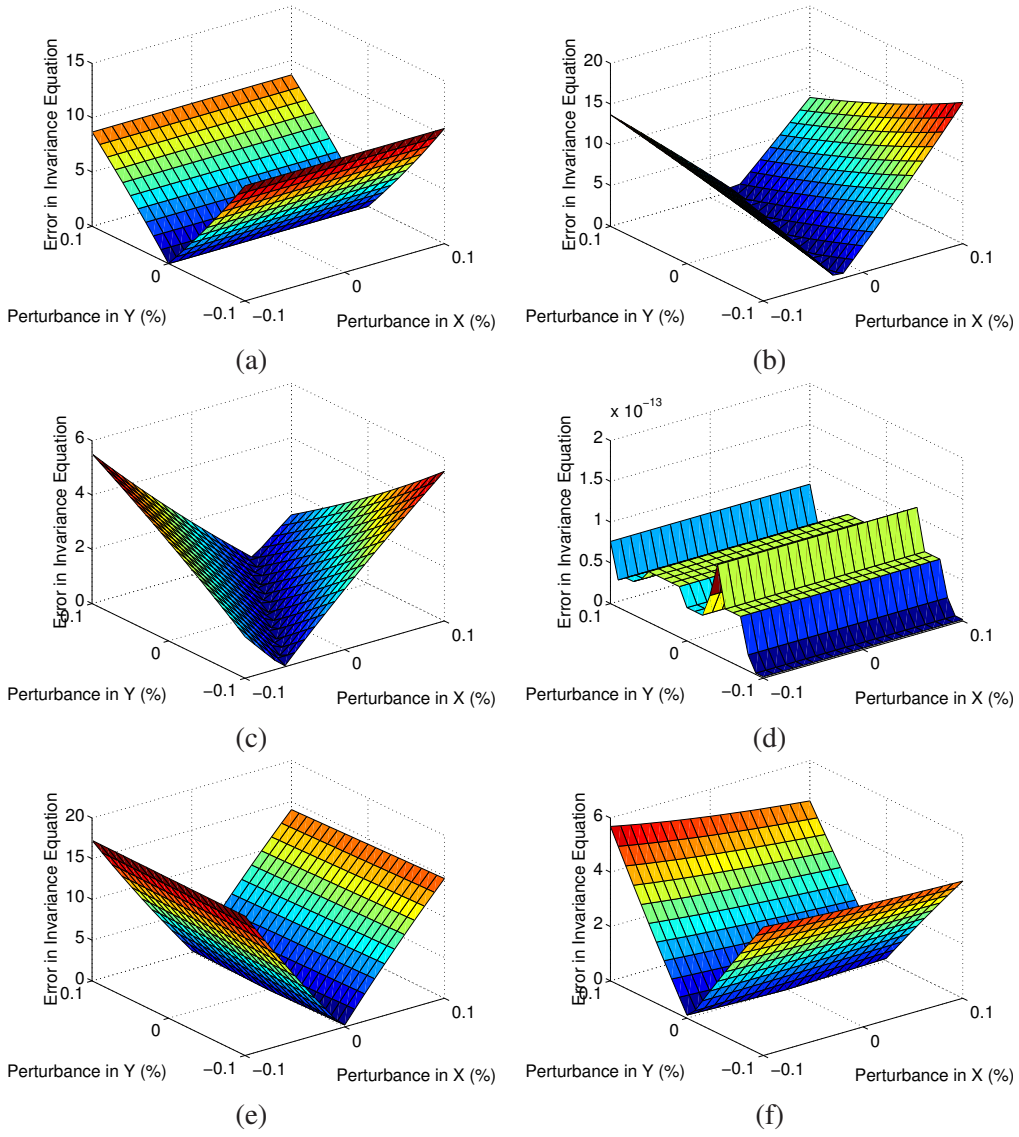


Figure 5.2: 3D-2D Projective Relation Error with respect to percentage of coordinate change in 2D projected coordinates of Table 5.3. (*a*) $\vec{x_1}$, (*b*) $\vec{x_2}$, (*c*) $\vec{x_3}$, (*d*) $\vec{x_4}$, (*e*) $\vec{x_5}$, (*f*) $\vec{x_6}$

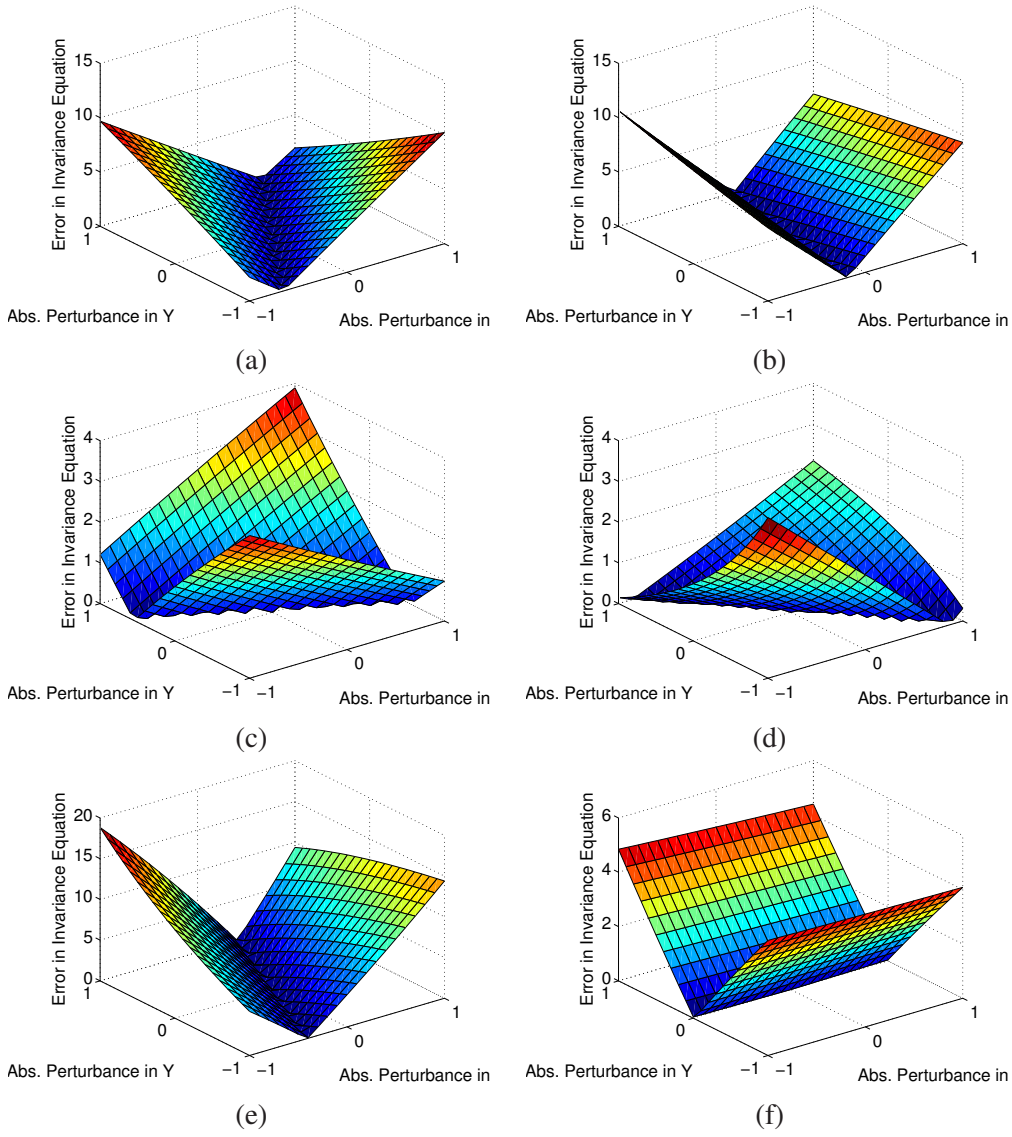


Figure 5.3: 3D-2D Projective Relation Error with respect to absolute coordinate change in 2D projected coordinates of Table 5.3. (*a*) $\vec{x_1}$, (*b*) $\vec{x_2}$, (*c*) $\vec{x_3}$, (*d*) $\vec{x_4}$, (*e*) $\vec{x_5}$, (*f*) $\vec{x_6}$

5.3.2 Robustness Analysis of Affine Invariants

In this second set of simulations, we experimented with affine invariants which are theoretically expected to be less prone to pixel errors. However, since these invariants are, by definition, not invariant against perspective projection, their usage should be in selective domains, where perspective distortion is limited. In order to test their robustness under orthographic and perspective projection cases, we organized three sets of experimental data. To create the first set, we projected a planar logo image on a quadric artificial surface that we defined. Then, we obtain both orthographic and perspective projections of this surface and mark the coordinates of 6 pre-selected points on these projections. As a result, we had 2 sets of 3D invariants obtained from 2 sets of points $\{1, 2, 3, 4, 5\}$ and $\{1, 2, 3, 4, 6\}$. For each of these two sets of points, we obtained 6 projections (3 point of views in orthographic and perspective mode) and check the errors in the invariance equations (Equations 5.11 and 5.12). A second set of data is obtained from a 3D Studio Max model of a Coke can. First, 3D coordinates of pre-selected points are marked on the model and recorded. Then, the model is rendered from three point of views with two types of camera (orthographic and perspective). The affine invariants are tested on these projections. Lastly, a real Coke can is captured in a real scene from three different views with two types of lenses. The first of these lenses is for simulating perspective distortion, whereas the second orthographic distortion.

5.3.2.1 Simulation I (2D Logo Projected on Artificial 3D Surface)

Base Images of logo projections are given in Figure 5.4. 6 points are marked on these base images, similar to the ones in Figure 5.5. Each pair of invariance equations, (5.11) and (5.12), obtained for 3D invariants by substituting 2D invariants obtained from a projection defines a line in invariant space. A pair of these lines that are obtained from two different views of the same 5 points is given in Figure 5.6. Invariant lines for two point sets from 6 different views (three orthographic and three perspective), whose 3D invariants are also plotted are given in Figure 5.7. The location of line intersections around the true 3D invariant is given in Figure 5.8. In this figure, green points denote intersections for corresponding point sets and red lines denote intersections by that occurred randomly. Note that green points are confined in a compact region that seems to be separable from the red ones.

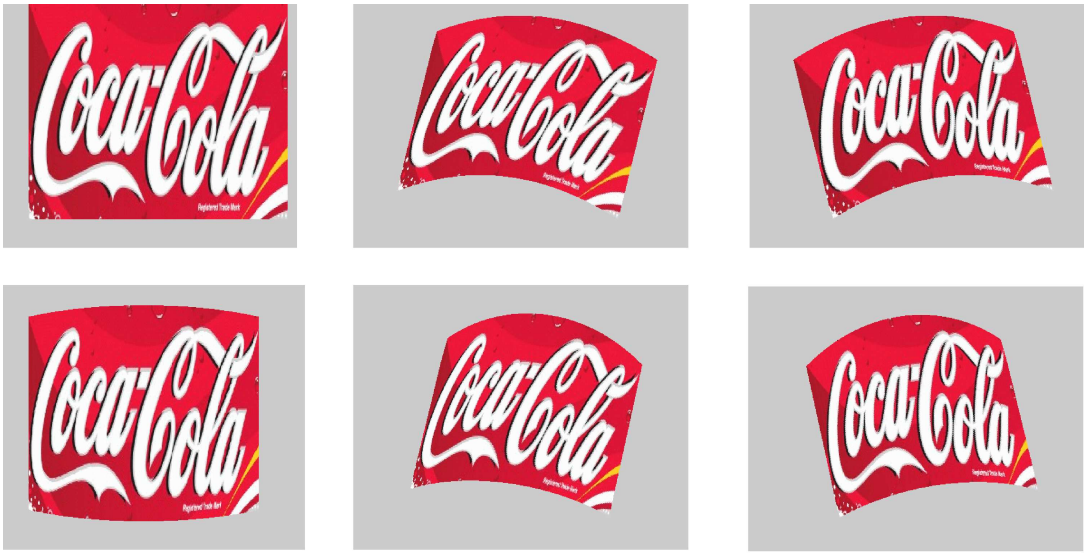


Figure 5.4: Images of projected logo obtained from different viewpoints with orthographic (upper row) and perspective (lower row) camera models.

(a) (b)

Figure 5.5: Marked points on (a) orthographic and (b) perspective images.

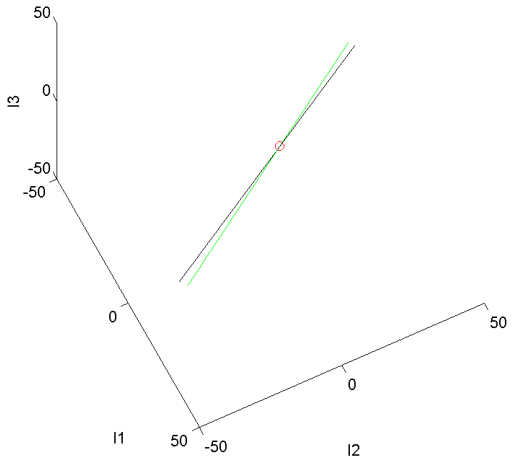


Figure 5.6: Invariant lines in (I_1, I_2, I_3) space, calculated from orthogonal and perspective projection of the same view.

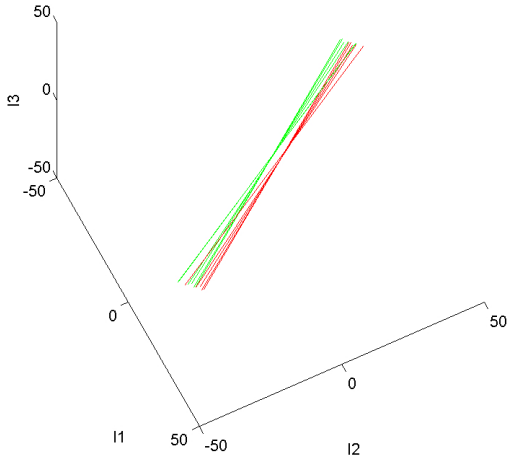


Figure 5.7: Invariant lines in (I_1, I_2, I_3) space, calculated for two different point sets from 6 different views including orthographic and perspective modes.

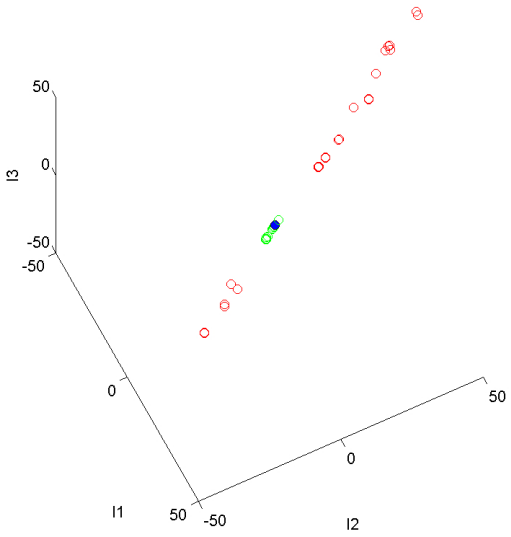


Figure 5.8: Distribution of correct (green) and erroneous (red) intersection points around 3D invariant point (blue).

5.3.2.2 Simulation II (3D Object Model - Artificial 2D Multi-view Rendering)

Base images that are used in a manner similar to the previous simulations are given in Figure 5.9. These images are obtained by rendering of 3D Studio Max models. Note that in these images, spheres that are manually embedded into the model to represent pre-selected points are also rendered. Hypothetical lines in 3D invariant space that result from 3D-2D invariance equations of two 5-point sets, viewed from three views with orthographic and perspective cameras are given in Figure 5.10. The location of line intersections around the true 3D invariant is given in Figure 5.11. In this figure, green points denote intersections for corresponding point sets and red lines denote intersections by chance. Again, green points are confined in a compact region that seems to be separable from the red ones.

5.3.2.3 Simulation III (3D Object Model - 2D Photos of Real Object)

Base images that are used in a similar manner as previous simulations are given in Figure 5.12. These images are obtained by a real camera with two different kinds of lenses (18mm for orthographic and 135mm for perspective).Note that pre-selected points are marked manually on these images. Hypothetical lines in 3D invariant space that result from 3D-2D invariance

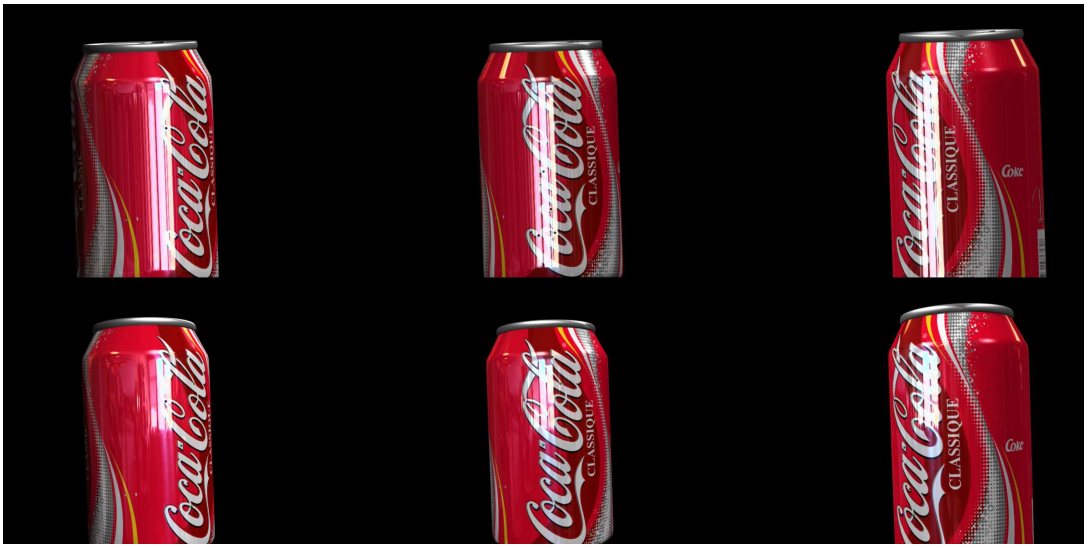


Figure 5.9: Marked points on orthographic (upper row) and perspective (lower row) projections of model Coke can from three different views.

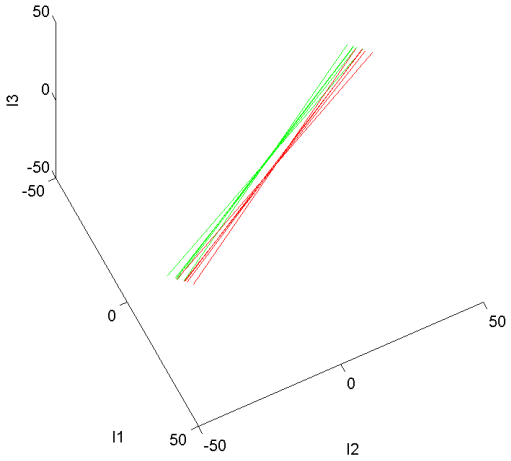


Figure 5.10: Invariant lines in (I_1, I_2, I_3) space, calculated for two different point set from 6 different views including orthographic and perspective modes.

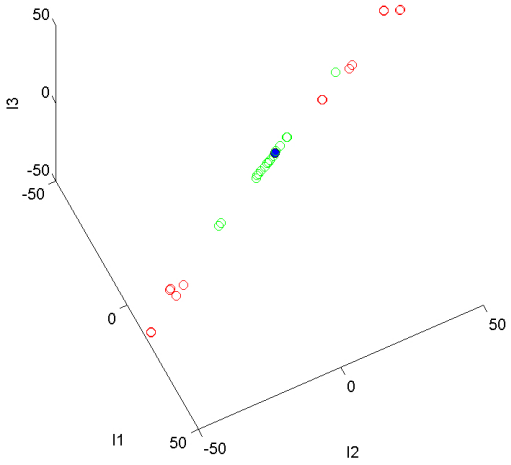


Figure 5.11: Distribution of correct (green) and erroneous (red) intersection points around 3D invariant location (blue) in (I_1, I_2, I_3) space.

equations of two 5-point sets, viewed from three views with orthographic and perspective cameras are given in Figure 5.13. The location of line intersections around the true 3D invariant is given in Figure 5.14. In this figure, green points denote intersections for corresponding point sets and red lines denote intersections by erroneously due to random behaviour. Again, green points are confined in a compact region that seems to be separable from the red ones. Together with simulations on the second dataset, these results proved the usability of affine invariance relations for matching 3D geometric structures.

5.3.2.4 Simulation IV (Correspondence Search between Multi-view Photographs of 3D Scenes)

The core element of object recognition algorithms is the matching algorithm, which in our case jointly utilizes invariants in two modalities, namely appearance and geometry. These two channels of discriminative information, i.e. local appearance descriptions and global geometric constraints on localization of local features, are utilized for detecting consistent groups of local feature matches between two local feature sets.

During the simulations of this section, validity of affine invariant relations are tested for ro-

Figure 5.12: Marked points on orthographic (upper row) and perspective (lower row) photographs of a real Coke can from three different views.

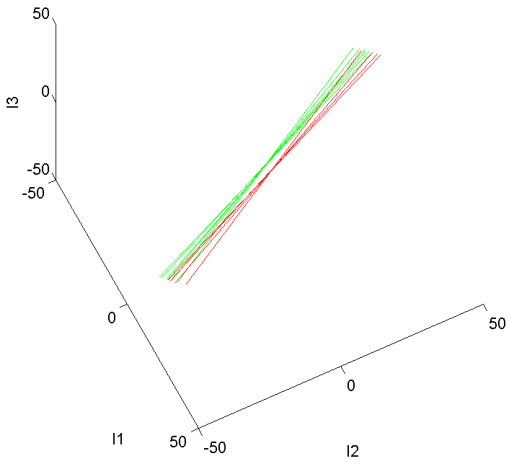


Figure 5.13: Invariant lines in (I_1, I_2, I_3) space, calculated for two different point set from 6 different views including orthographic and perspective modes.

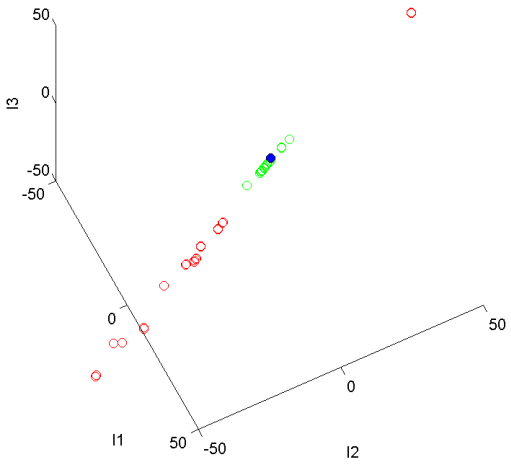
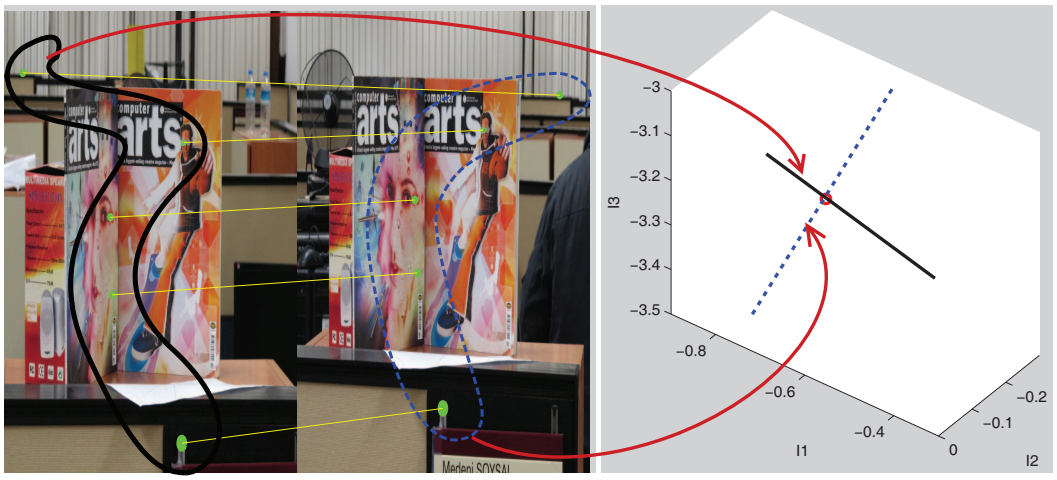


Figure 5.14: Distribution of correct (green) and erroneous (red) intersection points around 3D invariant location (blue).

bustness against perspective effects. In order to do so, local features are detected by DoG on different images of the same scene and then ground truth for matches between these images are constructed manually. The scene that is utilized during these experiments is deliberately set up in order to contain objects and structures which violate affine projection constraints given in Section 3.2.2. Two images of the aforementioned scene along with 5 ground truth matches are given in Figure 5.15a. The idea behind the geometric consistency step in our approach is utilizing relation in Equations (5.11) and (5.12) as a grouping constraint for appearance based matches [50, 51]. The implications of the geometrical meaning of these equations in the domain of 3D geometric invariants are worth emphasis at this point. Equations (5.11) and (5.12), each by itself represent a plane in 3D invariant space (I_1, I_2, I_3). The geometric entity that satisfy these two plane equations in a typical degenerate case is a line, which defines their intersection. In other words, each five point combination in an affine projection of a 3D scene to 2D imposes constraints on possible values of 3D invariants of the corresponding 3D scene points. These constraints limit the location of 3D invariants on a line, whose parameters are defined by 2D invariants (i_1, i_2, i_3, i_4) of the image points. Each projection of the same scene imposes the same type of constraints on 3D invariants of the real scene points. This result also means that 3D invariants of the scene must satisfy all of the line equations simultaneously, and therefore, reside at the intersection point of all these lines. This fact is



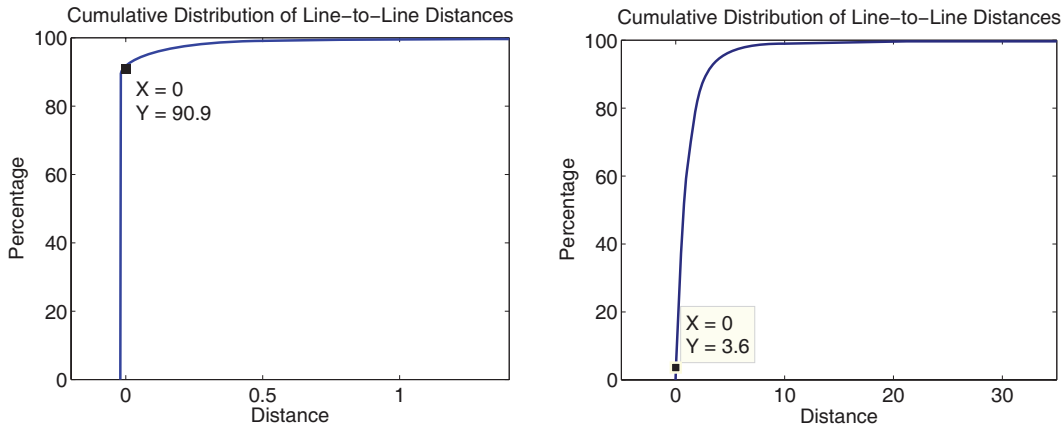
(a) (b)

Figure 5.15: (*a*) 5 local features repeatedly detected in two images of the same 3D scene, (*b*) The lines that imply the constraints on 3D invariant location. These lines are computed from 2D invariants calculated from coordinates of the 5 local features given in (*a*). (*Black straight line*) and (*blue dashed line*) are computed from the features on the (*left and right*) pictures, respectively. Their intersection, which is defined by 3D invariant coordinates of the scene points, is represented by the (*red circle*)

illustrated in Figure 5.15b for two images of the same scene.

The matching algorithm proposed in Section 5.4, exploits the aforementioned constraining relation for removing the ambiguity in the appearance-based matches. In parallel, 3D invariants of the consistent model groups are also computed using the line intersections in the invariant space. This enables us to check for consistency of 3D structure without explicitly computing a 3D model. In this last set of preliminary experiments, succeeding the robustness analysis in Sections 5.3.2.1, 5.3.2.2 and 5.3.2.3, we analyze the expected number of false positives arising from line intersections of random groups.

The results of these experiments indicate that random feature groupings between two irrelevant images may lead to lines that come significantly close to each other at some point (Figure 5.16) [51]. This information led us to the fact that, using invariant line intersections alone does not completely remove the ambiguity of matches. In order to overcome this fact, we performed the same analysis, but this time calculating the 3D invariant locations using two manually matched images. This way, we utilize the advantage of multiple images of the same group of local features for determining the precise 3D invariant location, and increase certainty in comparison to the previous single image case that can only narrow down the possible 3D invariant location to a line. In this setting, we select a reference image pair and two



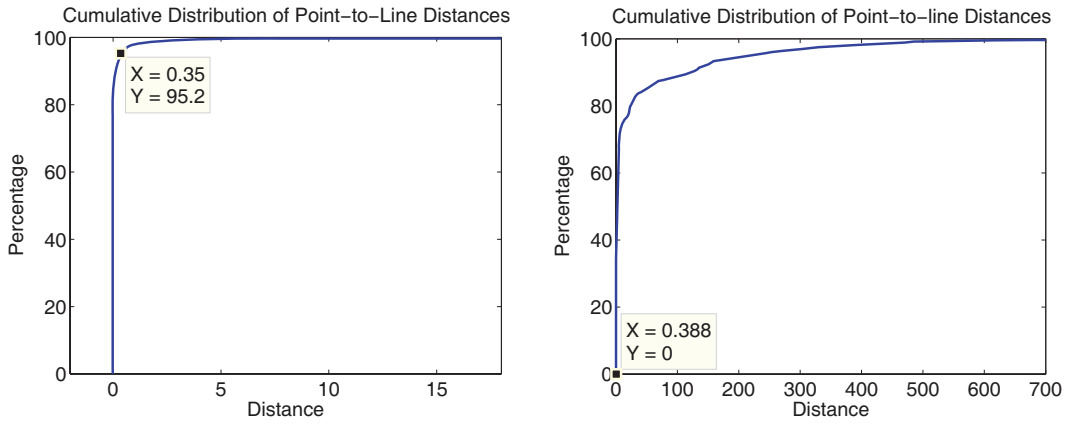
(a) (b)

Figure 5.16: (*a*) Line-to-line distances obtained from ground truth groupings. 90.9% of a total of 300 line pairs intersect each other. (*b*) Line-to-line distances (i.e. intersection of 2 lines) obtained from random generated groupings. 3.6% of 604,000 line pairs intersect each other.

test images, one being another view of the same scene and the other having totally irrelevant content. First, we generate a high number of random local feature matches between the reference images. The cumulative histogram of distances between lines induced by extremely high number of groupings of these random matches and the corresponding 3D invariant points that are computed from the reference image pair (Figure 5.17b). Next, for the test image with the same content, we compute cumulative histogram of distances between lines induced by ground truth match groups and the corresponding 3D invariant points that are computed from the reference image pair (Figure 5.17a). The results has shown that, although line-line intersections in the invariant space may lead to false positives, line-point intersections are very discriminative between true and false positives. Distribution of distances between the ground truth 3D invariants computed from two reference images and invariant lines defined by a third image is given in Figure 5.17.

5.4 A Method for 3D Object Recognition using Covariant Local Appearance Descriptors and Invariant Geometric Constraints

The proposed algorithm [51], which is designed considering the results of the preliminary experiments in Section 5.3, can be briefly summarized in three steps: In Step 1 of the algorithm, local appearance information is used to generate putative matches common between three images. During the modeling process, these three images are typically images of the



(a) (b)

Figure 5.17: (*a*) Point-to-line distances obtained from ground truth groupings. 95.2% of the 300 lines generated from groundtruth groupings has distance lower than 0.35 to their corresponding model points (*b*) Point-to-line (intersection of 2 lines to a third line) distances obtained from lines generated from random generated groupings. None of the 604,000 lines generated from random groupings has a distance lower than 0.388 to the model points

model scene taken, consecutively. During the testing step, third image is replaced with the test image. Step 2, performs similar to Random Sample Consensus (RANSAC) and searches for the 4 point match group that has the highest number of consistent matches in terms of 3D invariants. In Step 3, consistent match groups are used for testing other matches for consistency and expanding the group of geometrically consistent matches. This step is necessary, since, in real life photographs, affine assumption is violated in different amounts. Quadruplet groups containing grossly affected patches may miss other compatible matches, while less affected groups can successfully detect them.

The core matching procedure is explained in Algorithm 5.1. The matching method is utilized both in model library construction and recognition stages. The parameters of the algorithm typically varies between the two tasks. This variation is due to the nature of the scenes in training and test datasets. Experimental results for both of the tasks is presented in the following sections.

Algorithm 5.1 Proposed Invariant based Matching Algorithm

- thr_a: Appearance distance ratio threshold
- R: Number of iterations of the Selection/Verification Stage
- R_T: Number of iterations of the Expansion Stage
- thr_d: Distance threshold in the Selection/Verification Stage
- $g : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \xrightarrow{g} \mathbb{R}^4$: Mapping from 5 image coordinates to 2D invariants (i_1, i_2, i_3, i_4) defining lines in 3D invariant space
- $h : \mathbb{R}^4 \times \mathbb{R}^4 \xrightarrow{h} \mathbb{R}^3$: Mapping from 2 lines in 3D invariant space to the midpoint on the shortest path between them

A. Appearance-based Potential Match Selection

(1) Let f_{ij} represent j^{th} local feature location (x, y) and a_{ij} its description in the i^{th} image where $i = 1, \ldots, 3,\ j = 1, \ldots, N_i$.

(2) $M_{12} = \left\{ (p, q) : q = \arg\min_j \|a_{1p} - a_{2j}\| \wedge \frac{\|a_{1p} - a_{2k}\|}{\|a_{1p} - a_{2q}\|} > thr_a,\ \forall k \in [1, N_2] \setminus q \right\}$

(3) $M_{32} = \left\{ (r, q) : q = \arg\min_j \|a_{3r} - a_{2j}\| \wedge \frac{\|a_{3r} - a_{2k}\|}{\|a_{3r} - a_{2q}\|} > thr_a,\ \forall k \in [1, N_2] \setminus q \right\}$

(4) $C := \{\vec{c_i} = (c_{i1}, c_{i2}, c_{i3}) : (c_{i1}, c_{i2}) \in M_{12} \wedge (c_{i3}, c_{i2}) \in M_{32}\}$

Algorithm 5.1 Proposed Invariant based Matching Algorithm (cont'd)

B. Geometric Selection/Verification Procedure

Note that c_{ij} is the index of the feature from the j^{th} image in the i^{th} common feature triplet. Assume $c_{ij} = f_{jc_{ij}}$ from this point on for notational simplicity.

(1) **for** $i = 1 \rightarrow R$ **do**

Select 4 random matches from C, $S(i) = (\vec{c_p}, \vec{c_q}, \vec{c_r}, \vec{c_s})$

Let $S_k(i) = (c_{pk}, c_{qk}, c_{rk}, c_{sk})$ be features in the k^{th}*image*

for all $\vec{c_j} \in C \setminus S(i)$ **do**

$\vec{L_1} = g(S_1(i), c_{j1})$, $\vec{L_2} = g(S_2(i), c_{j2})$, $\vec{L_3} = g(S_3(i), c_{j3})$

$d_j = \|h(\vec{L_1}, \vec{L_2}) - h(\vec{L_3}, \vec{L_2})\|$

end for

$Z(i) \leftarrow \left\{\vec{c_p}, \vec{c_q}, \vec{c_r}, \vec{c_s}\right\} \cup T$, where $T = \left\{\vec{c_j} : d_j < thr_d\right\}$

end for

(2) **if** $\exists Z(i) : |Z(i)| > 4$ **then**

$B \leftarrow Z(\hat{i})$ where $\hat{i} = \underset{i \in \{1,\ldots,R\}}{\arg\max} |Z(i)|$

else

$B \leftarrow \emptyset$

end if

C. Expansion using Compatible Matches ($B \neq \emptyset$)

for $i = 1 \rightarrow R_T$ **do**

Select 4 random matches from B, $S(i) = (\vec{c_p}, \vec{c_q}, \vec{c_r}, \vec{c_s})$

Let $S_k(i) = (c_{pk}, c_{qk}, c_{rk}, c_{sk})$ be features in the k^{th}*image*

for all $\vec{c_j} \in C \setminus B$ **do**

$\vec{L_1} = g(S_1(i), c_{j1})$, $\vec{L_2} = g(S_2(i), c_{j2})$, $\vec{L_3} = g(S_3(i), c_{j3})$

$d_j = \|h(\vec{L_1}, \vec{L_2}) - h(\vec{L_3}, \vec{L_2})\|$

end for

$B \leftarrow B \cup T$ where $T := \left\{\vec{c_j} : d_j < thr_d\right\}$

end for

5.4.1 3D Object Model Library Construction from Images

Modeling process contains three main steps: (1) Construction of the adjacency graph among model images; (2) Iterative application of the core procedure defined in Algorithm 5.1 for identification of robust model features; (3) Creation of a model library using the results of step (2). In a typical case for a model image chain of 16 consecutive images, modeling process takes less than 2 minutes. The parameters of the procedure that are defined in Algorithm 5.1 are set to their nominal values of $thr_a = 1.5, thr_d = 0.03, R = 100$ and $R_T = 20$ during the modeling phase of the experiments in Section 5.4.3.

5.4.1.1 Construction of Model Image Adjacency Graph

As previously stated in Section 5.4, the core matching/verification part of the proposed algorithm operates on three adjacent model images. Therefore, the pairwise neighborhood information of the model images is a prerequisite for the modeling process. Although this information can be considered as an extra input, since most of the time images are taken successively in an ordered manner, the manual labor it costs is negligible. Pairwise neighborhood information of images are used to construct an undirected graph, whose nodes are model images that are connected by edges representing neighborhood relationships among them. Next, this graph is parsed to generate a list of unique paths with length two, in other words, all image triplets that are continuous in terms of viewing conditions are listed. In order to illustrate this process, let's assume we have five images $(I_1, I_2, I_3, I_4, I_5)$ taken successively using a camera rotating around an object. The pairwise neighborhood information for these images is represented by the set $A = \{(1,2),(2,3),(3,4),(4,5)\}$. The adjacency graph corresponding to this set of neighbors correspond to a simple chain: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$. The output of the adjacency graph creation step for this example is then a set G,which contains three triplets such that $G = \{(1,2,3),(2,3,4),(3,4,5)\}$.

5.4.1.2 Identification of Robust Model Features

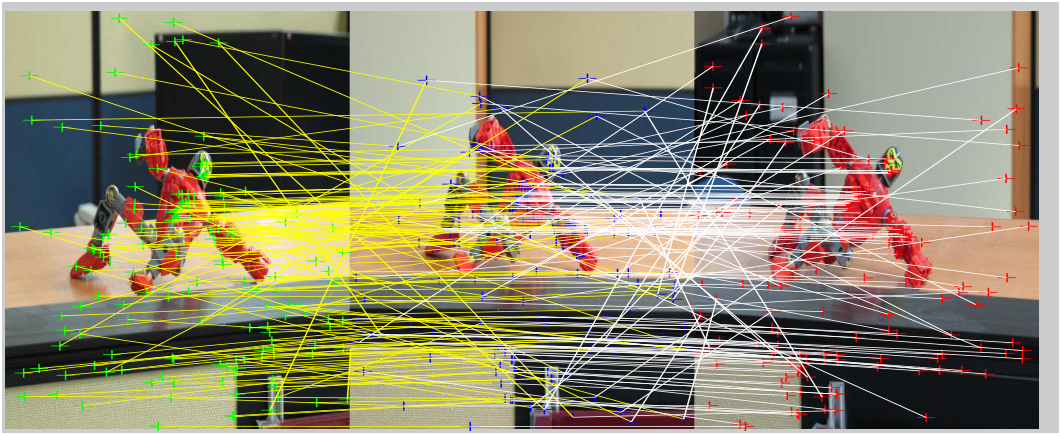
Each triplet in set G is considered as a source model features which are robust enough in terms of appearance such that they are detected in at least three model images. In order to reveal those model features that are compatible in terms of invariants of both appearance (Sec-

tion 2) and geometry (Section 3.3), the method defined in Algorithm 5.1 is utilized. During the processing of triplets, the first step is to determine initial feature correspondences which will constitute the input to the verification step. This is performed by using SIFT descriptors and an ambiguity threshold of thr_d on the second-to-best match distance ratios. It is observed during preliminary experiments that in images with multiple identical features, this simple elimination method discards a significant number of useful matches. For this reason, filtering the best matches is omitted, and instead a potential match list is used to facilitate the use of multiple candidates for each feature. This approach has proved fruitful in successful methods in the literature [176]. Despite these facts, we concentrate on the geometric verification process in this work and adopt this naive appearance-based filtering criterion in this initial version of our algorithm. After the pairwise appearance-based matching/filtering step between first-second and second-third images in the triplet, based on descriptors d_{ij} in steps A1 through A3 of Algorithm 5.1, in step A4, a set of common matches (C) among three images are determined.

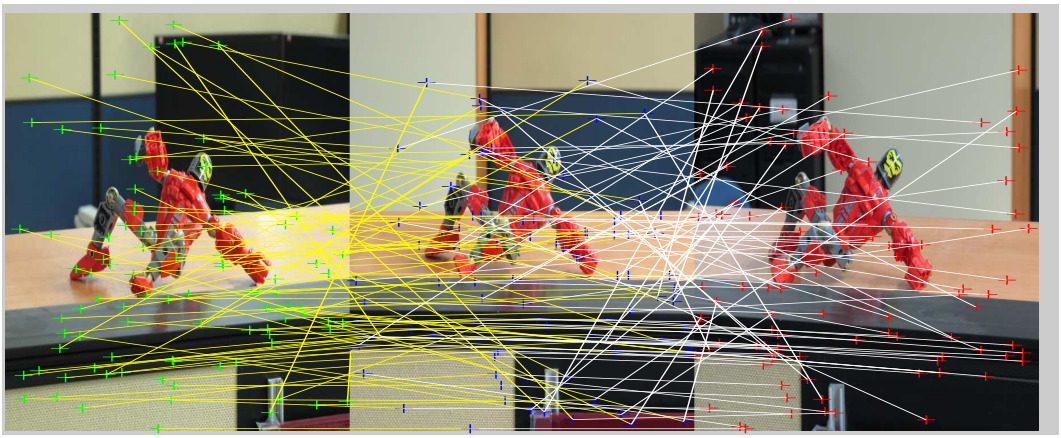
In Step B1, a predefined number (R) of quadruplet groups $S(i)$ from set C are selected in order to be tested as a basis. For this aim, one-by-one, each of the remaining matches $\vec{c_j}$ in set C is considered as the source of 5^{th} point for the computation of 3D invariant lines (Section 3.3). The midpoints of shortest paths, namely $h(\vec{L_1}, \vec{L_2})$ and $h(\vec{L_3}, \vec{L_2})$ from lines $\vec{L_1}$ and $\vec{L_3}$, which are defined using 2D coordinates of five features in the first and third images, to the line defined by the second image features $\vec{L_2}$ are computed for each selection of $(\vec{c_j})$. The distance between these midpoints, which is used as the measure of group geometric consistency, is computed as the Euclidean distance between midpoints $h(\vec{L_1}, \vec{L_2})$ and $h(\vec{L_3}, \vec{L_2})$. The best basis and its compatible matches are determined in Step B2 of Algorithm 5.1. If there is at least one basis with an additional compatible match, then Step C is performed for mining more matches among others. An example demonstrating the results obtained by this approach, under viewing conditions that can be considered marginal according to the weak perspective assumption, is presented in Figure 5.18.

5.4.1.3 Model Library Structure

The robust model features, which are detected among at least three images and compatible according to the 3D invariant constraints, are determined by the iterative application of Al-



(a)



(b)

(c)

Figure 5.18: Demonstration of the atomic operation in the modeling process under challenging imaging conditions: Three consecutive images (left, center and right) of a scene with complex 3D structure taken from a relatively short distance are input to Algorithm 5.1. (*a*) 165 appearance-based matches common among 3 images (*b*) 97 matches that are filtered out using invariant geometric constraints (*c*) 68 matches that are verified using invariant geometric constraints

gorithm 5.1 as explained in Section 5.4.1.2. As the final step of the modeling process, the output of the preceding steps should be organized in a model library structure that enables efficient utilization in the recognition process. The proposed model library contains two main elements: (1) Refined Model Feature List; (2) Model Match Graph.

Refined Model Feature List is populated by parsing the results obtained during the process defined in Section 5.4.1.2 , and recording each unique feature of each model image that is seen in at least one of the compatible groups (B) in Step C of Algorithm 5.1. The connections between these refined list of image features are formed considering every B set and are represented in a graph structure. In our implementation, this graph structure is represented as a sparse matrix, whose columns represent images, and rows represent verified robust local features. This data structure is identical in terms of its content and purpose to the one used in [176], and therefore will be called as *feature view matrix* similar to its counterpart.

5.4.2 3D Object Recognition from Images

Application of the proposed approach to recognition of objects in test images utilizes Algorithm 5.1 in a way similar to the modeling process defined in Section 5.4.1. The nuances specific to recognition are explained in the following sections. Briefly, the recognition process can be explained by three main steps: (1) Appearance based putative correspondence determination between the test image features and each of the refined model image feature sets; (2) Identification of Geometrically Consistent 3D Features via application of Algorithm 5.1 (3) Assessment of confidence level for the test image features that are matched in step (2).

5.4.2.1 Putative Correspondences using Local Appearance Descriptors

The first step in recognition is generating set of putative matches between the test image and each model image. This is performed in a per model image basis, comparing SIFT descriptors in the refined model image features list with the test image features and applying an ambiguity threshold of thr_a on the second-to-best match distance ratios.

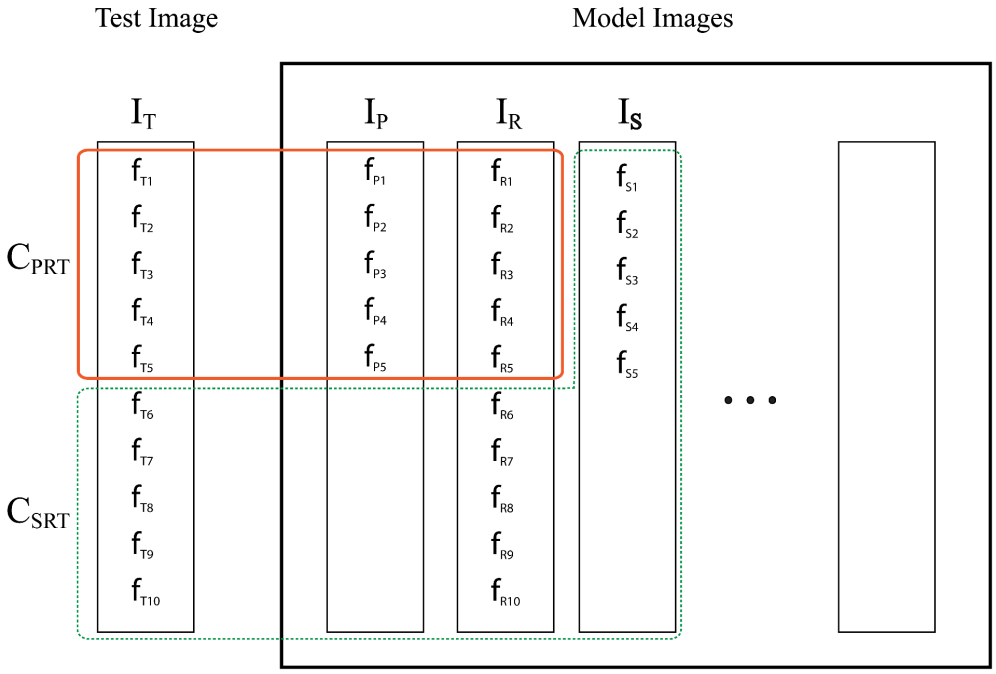


Figure 5.19: Illustration of the common feature selection process for recognition, which is defined in Section 5.4.2.2. Test image I_T is compared to model image I_R using appearance matches. According to the *feature view matrix*, features $f_{R1}, \ldots, f_{R5}$ in I_R are linked to features $f_{P1}, \ldots, f_{P5}$ in I_P, while features $f_{R6}, \ldots, f_{R10}$ in I_R are linked to features $f_{S1}, \ldots, f_{S5}$ in I_S. Matches between I_T and I_R are divided into two common match groups, namely C_{PRT} and C_{SRT} for geometric verification step (Step B) of the method described in Algorithm 5.1.

.

5.4.2.2 Iterative Identification of Geometrically Consistent 3D Features

In the modeling step (Section 5.4.1), the proposed method is applied on each model image triplet in the model image adjacency graph defined in Section 5.4.1.1. Triplets are used also in the recognition process, but they are selected in a different way. First of all, the third feature in any triplet is always used as a test feature. In addition, during the per model image basis comparison of a model library with a test image, the second feature in the triplet is the feature in the model image under consideration that matched to the third feature. The selection process of the third feature is illustrated by a visual example in Figure 5.19.

In the comprehensive example of Figure 5.19, we have a test image I_T and three model images I_P, I_R and I_S. During the appearance-based matching process, it is assumed that 10 features $(f_{T1}, \ldots, f_{T10})$ of model image I_T has been matched to features $(f_{R1}, \ldots, f_{R10})$ of the model image I_R under consideration. In order to select the third image to be used in the common feature triplets, *feature view matrix* defined in Section 5.4.1.3) is utilized. Features $f_{R1}, \ldots, f_{R5}$

are complemented by features $(f_{P1}, \ldots, f_{P5})$ in image I_P, and used for geometric verification of test features $f_{T1}, \ldots, f_{T5}$. On the other hand, features $f_{R5}, \ldots, f_{R10}$ in I_R are used together with their counterparts $f_{S1}, \ldots, f_{S5}$ in image I_S for evaluation of test features $f_{T5}, \ldots, f_{T10}$. The processing and evaluation of each of these groups are performed as described in Step (B) of the method described in Algorithm 5.1.

5.4.2.3 Metrics for Object Detection

Another nuance of the recognition process is the calculation of confidence measures for each test image feature that is matched to the model. This calculation is performed in synchronization with Step (C) of the method described in Algorithm 5.1 without adding extra complexity. The modified version of the third step for recognition is as follows:

Let $V(i) = 0$ where $i = 1, \ldots, |C|$
for $i = 1 \rightarrow R_T$ **do**
  Select 4 random matches from B, $S(i) = (\vec{c_p}, \vec{c_q}, \vec{c_r}, \vec{c_s})$
  Let $S_k(i) = (c_{pk}, c_{qk}, c_{rk}, c_{sk})$ be features in the k^{th}*image*
  for all $\vec{c_j} \in C \setminus S(i)$ **do**
    $\vec{L_1} = g(S_1(i), c_{j1})$, $\vec{L_2} = g(S_2(i), c_{j2})$, $\vec{L_3} = g(S_3(i), c_{j3})$
    $d_j = \|h(\vec{L_1}, \vec{L_2}) - h(\vec{L_3}, \vec{L_2})\|$
  end for
  $B \leftarrow B \cup T$, where $T = \left\{\vec{c_j} : d_j < thr_d\right\}$
  $V(j) \leftarrow V(j) + 1, \ \forall j \in \{x : d_x < thr_d\} \cup \{p, q, r, s\}$
end for

Each of the test image features c_{3j} are filtered according to threshold thr_v on confidence measure $V(j)$, which is actually the number of compatibility votes it receives from other compatible features. This can be interpreted for each test feature, as a measure of its coherence with the group of compatible features that it belongs.

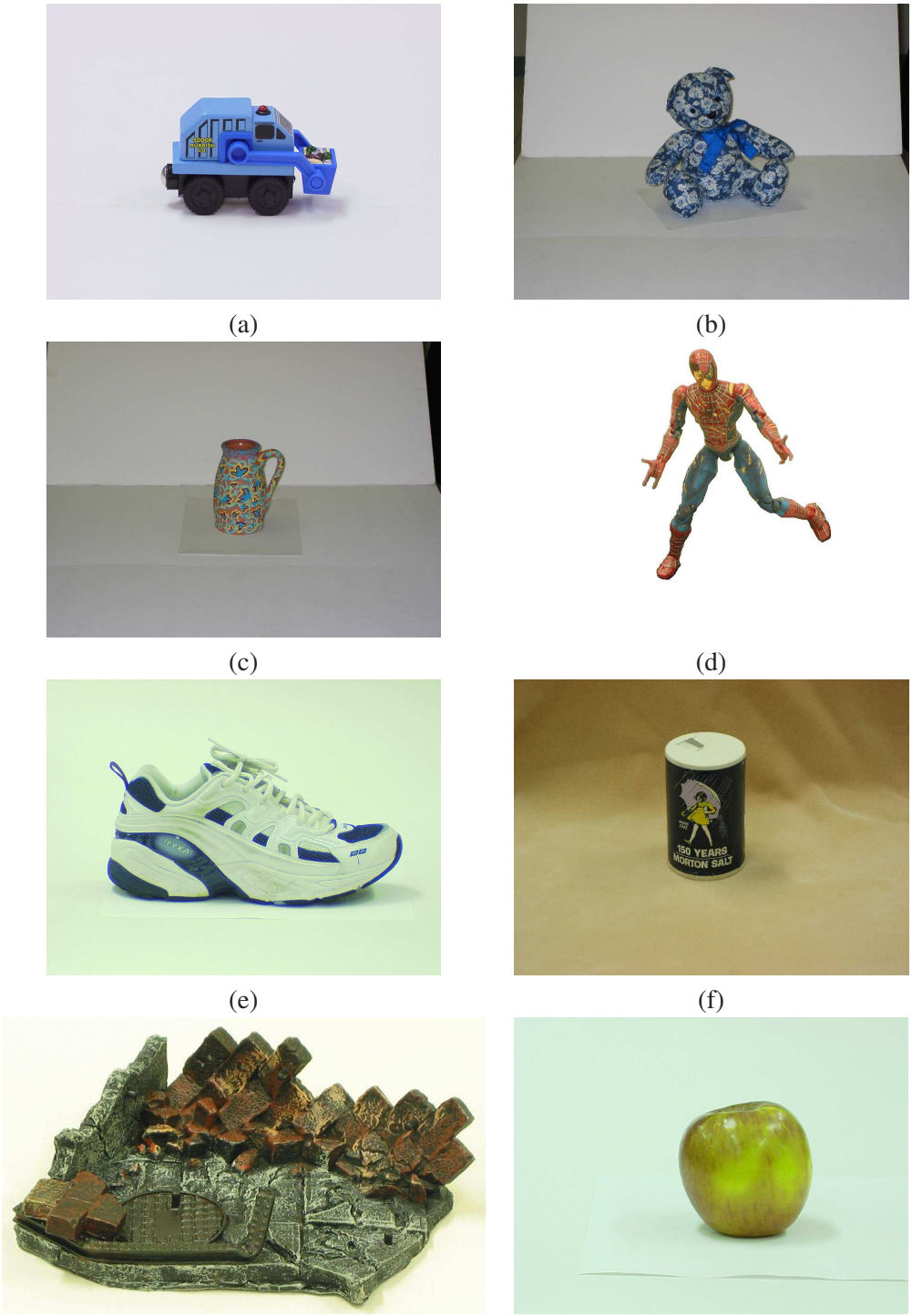
5.4.3 Experimental Results

We tested the proposed approach on a publicly available dataset [1], which has been created for comparative analysis of a diverse set of object recognition algorithms [176]. This dataset contains model images for eight different objects, which will be shortly referred to as *truck, bear, vase, spidy, shoe, salt, rubble* and *apple*. Each object is modeled using 16-20 images, except for the apple object, which has 29 images. Images used during the modeling process are generated in a controlled environment without any clutter (Figure 5.20). During the experiments, Difference-of-Gaussian (DoG) [75] and Harris-Affine [44] detectors are used for extraction of local features. In both cases, local appearance is characterized by the SIFT [75] descriptor.

Recognition experiments are performed on a seperate set of 51 images that contain heavy clutter, occlusion and photometric variations. Each test image contains instances of at least 1 and at most 5 model objects. In a typical case, comparing a single test image against a model library of 16 consecutive images is performed around 1 minute excluding the SIFT descriptor comparison. The parameters of the method described in Algorithm 5.1 are set to their nominal values of $thr_a = 1.5, thr_d = 0.05, R = 100, R_T = 20$ and $thr_v = 8$ during the recognition phase.

Recognition performance of the proposed algorithm is compared to other methods that are tested on the same dataset (all use Harris-Affine detector and SIFT, one additionally uses color), as well as Lowe's Hough-based method [75], in Table 5.4. None of the compared methods, including the proposed, yielded false positives in their reported parameter settings, and therefore, compared in terms of recall performances.

[1] This dataset is publicly available at `http://www-cvr.ai.uiuc.edu/ponce_grp/data`.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.20: Model images for objects in the dataset. (*a*) toy truck, (*b*) teddy bear, (*c*) vase, (*d*) spidy (Spiderman action figure), (*e*) shoe, (*f*) salt (salt can), (*g*) rubble (rubble-covered stand for Spiderman action figure) and (*h*) apple

Table 5.4: Object Recognition Performances (recall) on the Dataset used in [176]

	Truck	Bear	Vase	Spidy	Shoe	Salt	Rubble	Apple
Color (SIFT) [176]	12/12	11/11	12/12	4/4	7/9	10/10	9/9	8/11
B&W(SIFT) Greedy [176]	12/12	11/11	12/12	4/4	5/9	10/10	9/9	5/11
B&W(SIFT) Alignment [176]	12/12	10/11	12/12	4/4	4/9	10/10	9/9	5/11
B&W(SIFT) RANSAC [176]	9/12	11/11	11/12	3/4	2/9	9/10	8/9	3/11
Hough-based (Alg. 4.1) [75]	6/12	10/11	11/12	3/4	2/9	8/10	7/9	1/11
Proposed (DoG)	12/12	11/11	12/12	3/4	3/9	8/10	8/9	1/11
Proposed (Harris-Affine)	12/12	11/11	12/12	4/4	6/9	10/10	9/9	1/11

The performance of the DoG variant is much lower than its counterparts, except the Hough-based method that depends heavily on the planarity assumption. These changes affect the localization of the DoG detector negatively, leading to a reduced appearance-based matching performance. Examples of successful recognition results for the DoG variant of the proposed method is provided in Fig. 5.21. On the other hand, examples of test cases, where DoG variant of the proposed method fails, are given in Fig. 5.22.

The Harris-Affine variant of our approach, however, performed superior than most of compared methods in most of the cases (Table 5.4), despite its simplicity and naive appearance-based matching step. Substituting the Harris-Affine detector in place of DoG during the local feature detection step had a significant positive effect on the performance of the proposed method. Examples of successful recognition results for the Harris-Affine variant of the proposed method is provided in Fig. 5.23. Misdetections are reduced to 3 instances for the *shoe* object, and completely removed for the *spidy, salt* and *rubble* objects (Fig. 5.24). Still, the problems for the *apple* object persists, since they are mostly related to the low discriminative power of the local descriptors. Local features that are detected on the apple object can not be uniquely identified by their SIFT descriptions. Considering this observation, the stringent ambiguity threshold ($thr_a = 1.5$) utilized in the first step can be argued as the main cause of early feature elimination in *apple* object tests. The compared methods handle this problem by enabling many-to-one matching between model and test image local features and omitting the ambiguity threshold. This problem will be further elaborated in the conclusions section.

It is worth mentioning that the proposed method achieved this performance with a much

Figure 5.21: Representative examples of successful recognition results for *truck, bear, vase, spidy, salt, rubble* and *shoe* objects (top to bottom) using the DoG variant of the proposed method (Alg. 5.1)

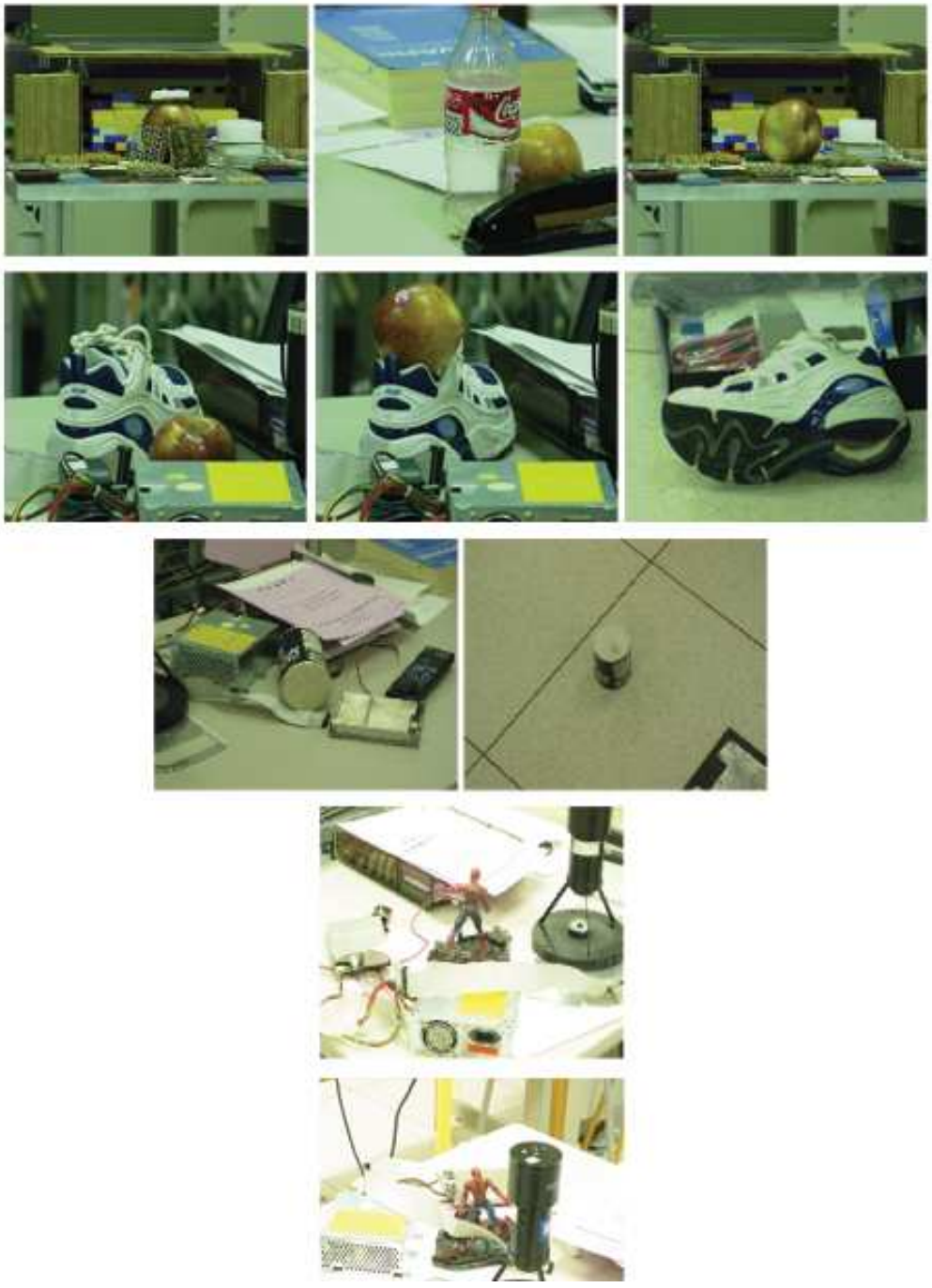


Figure 5.22: Examples of test cases, where recognition fails for *apple, shoe, salt, spidy* and *rubble* objects (top to bottom) using the DoG variant of the proposed method (Alg. 5.1).

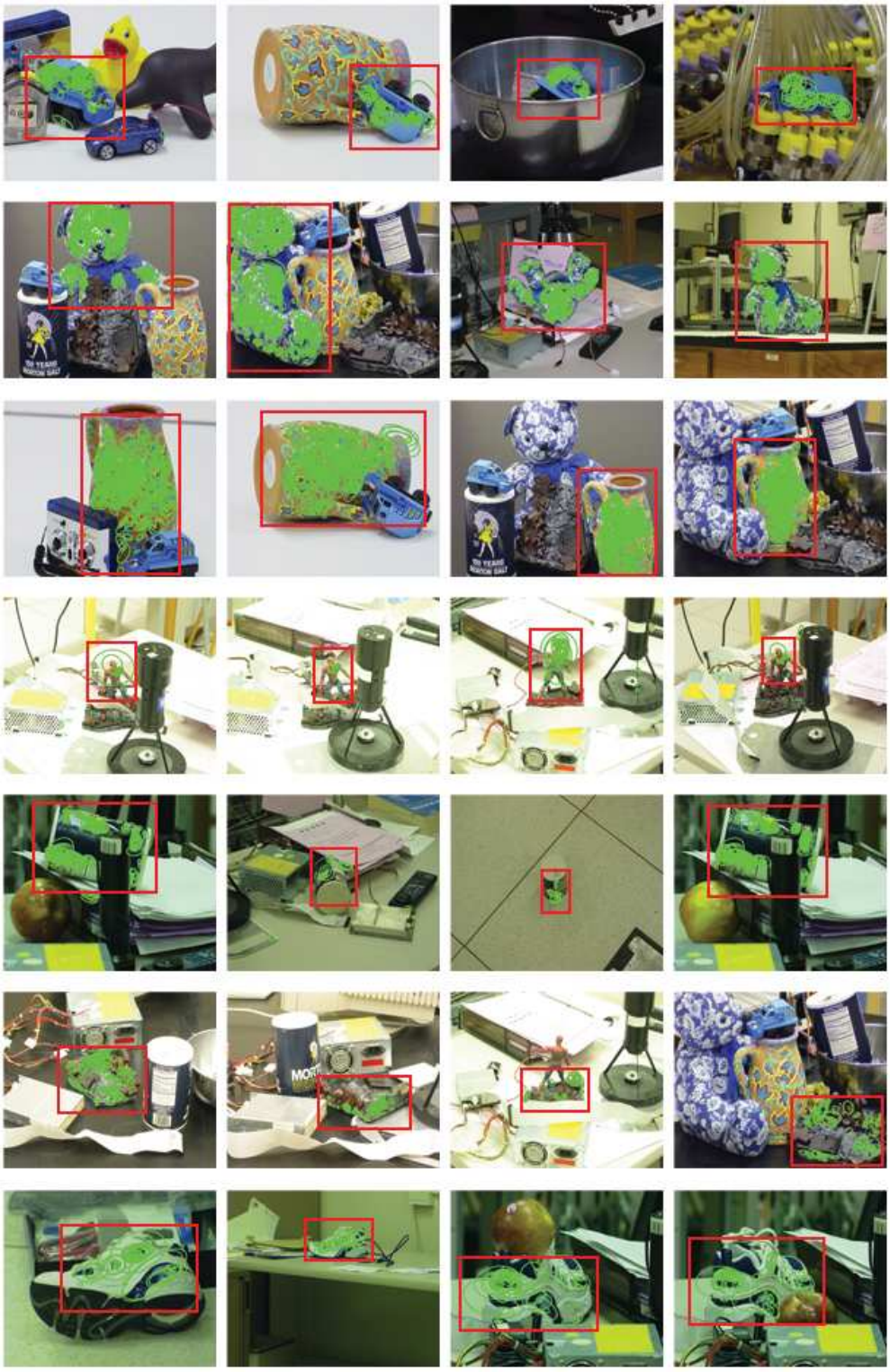


Figure 5.23: Representative examples of successful recognition results for *truck, bear, vase, spidy, salt, rubble* and *shoe* objects (top to bottom) using the Harris-Affine variant of the proposed method (Alg. 5.1)

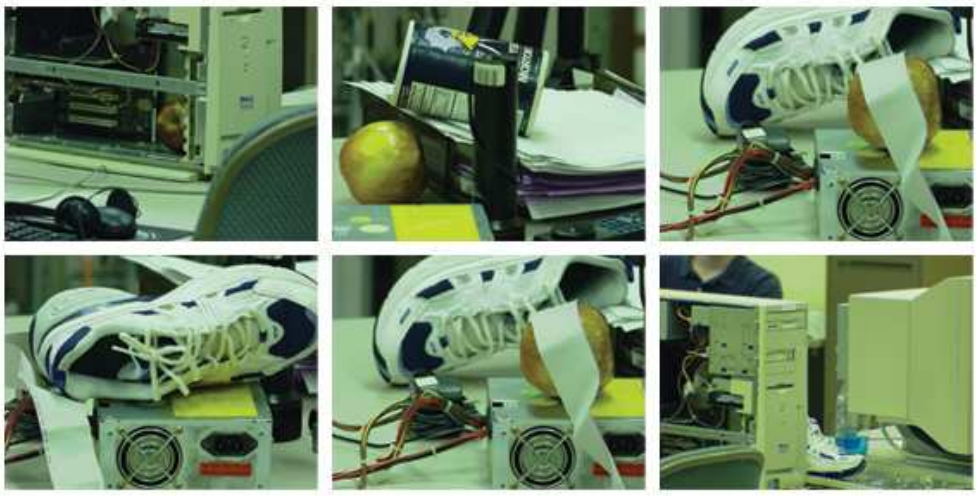


Figure 5.24: All three cases, where recognition fails for *shoe* (bottom row) and representative examples of cases, where recognition fails for *apple* (top row) objects using the Harris-Affine variant of the proposed method (Alg. 5.1).

lower modeling complexity (2 minutes in Matlab vs. 30-70 hours in C++). On the other hand, our method performs recognition in comparable times with its counterparts, although our time computations are performed using a Matlab implementation, in contrast to C++ implementations of the compared methods [176]. But this comparison can not be considered as fair, since computations are performed on different PC configurations.

5.5 Discussion and Future Work

In this chapter, invariant geometric relations between invariants of 3D-3D and 2D-2D are investigated. These relations that are designed to bridge the gap experienced in 3D-2D projection that results from the loss of depth information are analyzed in terms of applicability in practical cases. In order to perform this evaluation, simulations on controlled datasets have been conducted and important results have been recorded. The first of these results is the fragility of the relations that are defined for the general perspective projections. The second important result is the robustness of the relations that are defined for the affine projection approximations on appropriate projection conditions that are artificially created. The last and the most important result is the extended robustness observed for affine relations even under cases where perspective effects become significant in projections.

Affine relations between geometric invariance, which emerge as a promising alternative for

extending the model-based object recognition framework to 3D objects were selected for extensive experimentation in the rest of this chapter. In the light of the preliminary experiments in Section 5.3, we developed a method that utilize local feature based appearance from images and geometric invariance relations among these features for finding correspondences between the object model and real life images of an object.

The method that we presented in Section 5.4, combines local features that are appearance-based invariants and 3D affine geometric invariants for 3D object recognition [51]. Utilization of 3D geometric invariants in combination with distinctive and robust local representation of appearance has brought us two significant advantages: (1) The *grouping problem* that prevented widespread use of the powerful 3D geometric invariants [149] is ameliorated by narrowing down the search space using discriminative power of local feature descriptors; (2) Limited discriminative power of local descriptions is complemented by 3D geometric invariants at the early steps of recognition as a substitute for explicit modeling of the 3D object, which is costly both to construct and use. Our experiments provided strong evidence that the adopted approach efficiently utilizes multi-view model images during modeling and recognition phases, leading to performance comparable with variants of a more complex method [176].

The proposed method is compared both in terms of its components (Table 5.1) and recognition performance (Table 5.4) to its counterparts. This comparison is suggestive of the promise of the proposed method after proper supplementary tools are integrated at its early stages. One of these tools is the relaxation of the appearance-based matching step, lifting the 1-1 correspondence constraint between local appearance descriptors. Utilization of a simple color descriptor has also been reported to have a positive effect on the performance [176], and therefore, will be considered in the future work. Testing the method with other alternative appearance descriptions, optimizing the parameters/structure of the recognition step and reducing the computation time using simple grouping constraints [75] are the next steps in our agenda.

CHAPTER 6

CONCLUSIONS

In this final chapter of this dissertation, we recapitulate the contributions of our research and discuss possible directions for future work in the light of the insight gained from the conducted experimentation and research.

6.1 Summary and Conclusions

In this dissertation, we have considered a variety of frameworks for utilizing geometric invariants in synchronization with local features for the purpose of object recognition. In this way, we aim to exploit the solid mathematical foundations of geometric invariance for incorporating the strong spatial constraints between relatively small groups of local features among significant clutter. The planar object recognition part of our research has progressed from a hybrid method for matching local features in a model image to the ones in a query image, which iteratively evaluates the nearest appearance descriptor based potential matches, to a semi-local neighborhood model, which utilizes joint appearance and geometry based descriptions, and finally to a global model, which utilizes joint appearance and geometry descriptions of local features without any scale restriction on the neighborhood. Next, relations derived as part of the geometric invariance literature in the past are analyzed for their adaptability to 3D object recognition and important experimental findings are highlighted.

A Hybrid Method for Robust Correspondence Search Between Partially Planar Objects. Section 4.1 presented the application of hybrid representations of local features in terms of local appearance and semi-local geometry to one-to-one image matching. The use of affine 2D geometric invariants, namely barycentric coordinates, enabled us to perform robust detection

of correspondences between semantically related parts of the images. This chapter presented a hierarchy of simulations on a constrained dataset containing partially planar objects. Experimental results presented in Section 4.1.3 show the robustness of the joint representation and the matching method against appearance variations, which severely degrade the performance of local appearance descriptions extracted from small patches. Although, this method is conceptually primitive, its performance on a constrained dataset motivated future research [48].

During the experiments, it is observed that making a premature matching decision based on solely appearance leads to significant drop in the number of true matches, and therefore, the overall performance. On the other hand, the experiments showed that utilization of the hybrid description, which is constructed by combining appearance and geometric information, lead to a higher matching performance.

A Framework for Joint Utilization of Vector Quantized Appearance and Geometry. Section 4.2, has considered a method for extending the nearest appearance descriptor based matching scheme of Section 4.1 by adapting the method to utilize vector quantized local appearance descriptions. This extension provided a greater degree of robustness against appearance variations, which becomes especially important when searching for objects in real life scenes. In addition, hybrid descriptions that are based on multiple small groups of points, quads, are introduced. Lastly, the method is applied to a realistic unconstrained dataset that is created for natural scene logo detection. In the experiments of Section 4.2.3, the robustness of the proposed method in template matching within datasets with harsh appearance variations.

In the experiments of Section 4.1, it was shown that utilization of the hybrid description, which is constructed by combining appearance and geometric information, lead to a higher matching performance. In the method presented in this chapter, appearance-based similarity step is modified in order to render the solution less dependent on the context. During the scene logo detection experiments, it is observed that performing appearance matching based on more robust appearance descriptions that are quantized using k-means, can lead to successful results in real life applications. On the other hand, handling the grouping problem during the computation of geometric descriptors by considering only a predefined number of nearest spatial neighbors can be argued as a reason behind the degraded performance that is observed under extreme scale changes.

An Extended Framework for Joint Utilization of Vector Quantized Appearance, Geom-

etry and Significance-based Grouping. Section 4.3, has presented an evolved version of the method described in Section 4.2. This method, inherited the strong properties of the previous, and augments it with a novel scheme for ameliorating the grouping problem in high clutter and large changes in scale. This method has been evaluated on a dataset that is much larger than the one used in Section 4.2, and the result proved that the method work with significant confidence under harsh real life conditions.

In the experiments of Section 4.2, handling the grouping problem during the computation of geometric descriptors by considering only a predefined number of nearest spatial neighbors is identified as a source of the degraded performance that is observed under extreme scale changes. In the light of this observation, the method proposed in Section 4.3 is equipped with the means that render the local feature grouping process independent of the location of the local features. During the experiments performed using this modified version of the method, it is observed that successful results are obtained even in challenging conditions of a realistic scene logo recognition dataset.

Joint Utilization of Appearance and Geometry for 3D Object Recognition. Chapter 5 have presented some important results from the literature [149, 150], which investigates the use of geometric invariants for 3D object recognition. Useful relations that have been derived in [149], which may enable the use of geometry in 3D object recognition from 2D images despite the lack of invariants under 3D to 2D projection [36] have been analyzed in detail. Simulations that have been performed on various projection scenarios, for assessment of practical applicability of these relations provide important results. Affine geometric invariance relations have proved quite robust in realistic simulations, in contrast with projective relations. In addition, during the experiments performed on both artificial and real data, it is observed that affine geometric invariance relations are robust against perspective effects up to an adequate level. This encourages the use of affine invariants instead of the more general projective invariants in cases where the projection parameters permit affine approximation.

Affine relations between geometric invariants, which emerge as a promising alternative for extending the model-based object recognition framework to 3D objects, were selected for extensive experimentation in the rest of Chapter 5. In the light of the preliminary experiments in Section 5.3, a method, which utilize local feature based appearance from images and geometric invariance relations among these features for finding correspondences between the

object model and real life images of an object, was developed. Two variants of the proposed method is compared both in terms of its components (Table 5.1) and recognition performance (Table 5.4) to its counterparts. Despite being in initial implementation state, the proposed algorithm has been proved to have much more to promise after integration of proper supplementary tools.

The experimental results of Section 5.4 has proved that the proposed method can perform better than its counterparts in the literature, especially in situations, where localization and repeatability performance of local features do not limit the performance. During the experiments, it is also observed that performance of local detectors, which are invariant against only 2D similarity transforms (such as DoG) can significantly degrade under significant viewpoint changes. Supporting this observation, the variant of the proposed method, which utilizes a local feature detector (Harris-Affine) that uses affine adaptation during feature localization performed significantly superior than the DoG variant under challenging viewing conditions.

An important factor limiting the performance of the proposed method under special cases is identified as the texture characteristics of the model object (such as *apple*), which in practice may lead to local feature patches that are indistinguishable in terms of appearance. It is concluded that this problem can be solved with the incorporation of a more robust appearance-based matching step.

6.2 Future Work

In the future, we will work towards extending the use of 3D to 2D projective invariant relations to practical instance-level object recognition scenarios. In addition, the use of 2D projective invariants in place of 2D affine invariants will be investigated in planar object recognition problem domain.

Planar Object Recognition with Joint Utilization of Appearance and Projective Geometry. Section 4.3 presented a mature method that is based on affine 2D invariants, which are also called barycentric coordinates. As discussed in Section 3.2, affine approximation is valid only when certain conditions are met. In order to extend the applicability of the method in Section 4.3, utilizing 2D projective invariants are planned.

Joint Utilization of Appearance and Geometry for 3D Object Recognition. A novel method for 3D object recognition in cluttered scenes via 3D geometric invariants was presented in Chapter 5. Despite the promising results obtained in experiments of Section 5.4, the proposed method has many improvement opportunities. The first of these tools is relaxation of the appearance-based matching step, lifting the 1-1 correspondence constraint between local appearance descriptors. Utilization of a more distinctive local description such as a simple color descriptor [176] appended to the original SIFT descriptor will also be considered in the future work. Testing the method with alternative appearance descriptions, incorporating fundamental matrix (Chapter 9 in [95]) as a global measure of consistency in the model verification step, optimizing the parameters/structure of the recognition step (Section 5.4.2) and reducing the computation time using simple grouping constraints [75] are the next steps in our agenda.

REFERENCES

[1] J. Mundy, "Object recognition in the geometric era: A retrospective," *Toward Category-Level Object Recognition*, pp. 3–28, 2006.

[2] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, pp. 179–187, Feb. 1962.

[3] L. G. Roberts, *Machine perception of three-dimensional solids*, pp. 159–197. MIT Press, 1963.

[4] A. Guzmán, "Decomposition of a visual scene into three-dimensional bodies," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 291–304, ACM, 1968.

[5] D. Waltz, "Understanding line drawings of scenes with shadows," in *The psychology of computer vision*, Citeseer, 1975.

[6] K. Sugihara and Others, *Machine interpretation of line drawings*, vol. 1. Citeseer, 1986.

[7] F. Ulupinar and R. Nevatia, "Shape from contour: straight homogeneous generalized cylinders and constant cross section generalized cylinders," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, pp. 120–135, Feb. 1995.

[8] M. Zerroug and R. Nevatia, "From an intensity image to 3-D segmented descriptions," in *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, pp. 108 –113 vol.1, Oct. 1994.

[9] O. Firschein and T. Strat, *RADIUS: Image understanding for imagery intelligence*. San Francisco CA: Morgan Kaufmann, 1997.

[10] A. Roland and P. Shiman, *Strategic computing: DARPA and the quest for machine intelligence, 1983-1993*. The MIT Press, 2002.

[11] S. Underwood and C. Coates, "Visual Learning from Multiple Views," *IEEE Transactions on Computers*, vol. C-24, pp. 651–661, June 1975.

[12] C. Goad, "Special purpose automatic programming for 3D model-based vision," in *Readings in computer vision: issues, problems,* (M. A. Fischler and O. Firschein, eds.), ch. Special pu, pp. 371–381, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987.

[13] O. Faugeras, J. Mundy, N. Ahuja, C. Dyer, A. Pentland, R. Jain, K. Ikeuchi, and K. Bowyer, "Why aspect graphs are not (yet) practical for computer vision," *CVGIP: Image Underst.*, vol. 55, pp. 212–218, Mar. 1992.

[14] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *The International Journal of Robotics Research*, vol. 3, p. 3, Aug. 1984.

[15] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Norwell, MA, USA: Kluwer Academic Publishers, 1985.

[16] N. Ayache and O. D. Faugeras, "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 1, pp. 44–54, 1986.

[17] R. C. Bolles and R. A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," *The International Journal of Robotics Research*, vol. 1, no. 3, pp. 57–82, 1982.

[18] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial intelligence*, vol. 31, pp. 355–395, 1987.

[19] A. Kalvin, E. Schonberg, J. T. Schwartz, and M. Sharir, "Two-dimensional, model-based, boundary matching using footprints," *Int. J. Rob. Res.*, vol. 5, pp. 38–55, Dec. 1986.

[20] D. Gavrila and F. C. A. Groen, "3D object recognition from 2D images using geometric hashing," *Pattern Recogn. Lett.*, vol. 13, pp. 263–278, Apr. 1992.

[21] K. Ikeuchi and T. Kanade, "Applying Sensor Models To Automatic Generation Of Object Recognition Programs," in *Computer Vision., Second International Conference on*, pp. 228–237, Dec. 1988.

[22] W. E. L. Grimson, *Object Recognition by Computer*. Cambridge, MA, USA: MIT Press, 1991.

[23] D. Huttenlocher and S. Ullman, "Object recognition using alignment," in *Proc. ICCV*, pp. 102–111, 1987.

[24] D. Thompson and J. Mundy, "Three-dimensional model matching from an unconstrained viewpoint," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 208–220, IEEE, 1987.

[25] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[26] G. Stockman, "Object recognition and localization via pose clustering," *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 3, pp. 361–387, 1987.

[27] J. Mundy and A. Heller, "The evolution and testing of a model-based object recognition system," in *Computer Vision, 1990. Proceedings, Third International Conference on*, pp. 268–282, Dec. 1990.

[28] S. Vinther and R. Cipolla, "Towards 3D object model acquisition and recognition using 3D affine invariants," in *Proc. 4th British Machine Vision Conf*, 1993.

[29] L. Du, G. D. Sullivan, and K. D. Baker, "3D grouping by viewpoint consistency ascent," *Image Vision Comput.*, vol. 10, pp. 301–307, June 1992.

[30] L. Du, G. D. Sullivan, and K. D. Baker, "Quantitative analysis of the viewpoint consistency constraint in model-based vision," in *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pp. 632–639, May 1993.

[31] S. Carlsson, "Projectively Invariant Decomposition and Recognition of Planar Shapes," in *Proc. 4th ICCV*, pp. 471–475, 1996.

[32] C. Rothwell, A. Zisserman, D. A. Forsyth, and J. Mundy, "Planar object recognition using projective shape representation," *International Journal of Computer Vision*, vol. 16, pp. 57–99, Sept. 1995.

[33] D. Liebowitz and A. Zisserman, "Metric rectification for perspective images of planes," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, pp. 482–488, IEEE Comput. Soc, 1998.

[34] S. Startchik, C. Rauber, and T. Pun, "Recognition Of Planar Objects Over Complex Backgrounds Using Line Invariants And Relevance Measures," in *Workshop on Geometric Modeling & Invariants for Computer Vision*, pp. 301–307, 1995.

[35] F. Klein, "A comparative review of recent researches in geometry," *ArXiv e-prints*, July 2008.

[36] J. Burns, R. Weiss, and E. Riseman, "The non-existence of general-case view-invariants," *Geometric invariance in computer vision*, pp. 120–131, 1992.

[37] C. Rothwell, D. A. Forsyth, A. Zisserman, and J. Mundy, "Extracting projective structure from single perspective views of 3D point sets," *1993 (4th) International Conference on Computer Vision*, pp. 573–582, 1993.

[38] I. Weiss and M. Ray, "Model-based recognition of 3D objects from single images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 116–128, 2001.

[39] H. Murase, "Visual learning and recognition of 3-D objects from appearance," in *International journal of computer vision*, pp. 39–50, June 1995.

[40] C. Schmid, P. Bobet, B. Lamiroy, and R. Mohr, "An image oriented CAD approach," in *Object Representation in Computer Vision II* (J. Ponce, A. Zisserman, and M. Hebert, eds.), vol. 1144 of *Lecture Notes in Computer Science*, pp. 221–245, Springer Berlin / Heidelberg, 1996.

[41] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 530–535, May 1997.

[42] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, pp. 224–270, 1994.

[43] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[44] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, "A Comparison of Affine Region Detectors," *International Journal of Computer Vision*, vol. 65, pp. 43–72, Oct. 2005.

[45] P. Moreels and P. Perona, "Evaluation of Features Detectors and Descriptors based on 3D Objects," *International Journal of Computer Vision*, vol. 73, pp. 263–284, Sept. 2006.

[46] K. Mikolajczyk and C. Schmid, "A Performance evaluation of local descriptors.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, pp. 1615–30, Oct. 2005.

[47] M. Soysal, A. A. Alatan, and T. Karadeniz, "Joint utilization of appearance and geometry for determining correspondences," *2009 24th International Symposium on Computer and Information Sciences*, pp. 60–65, Sept. 2009.

[48] M. Soysal and A. A. Alatan, "Joint Utilization of Appearance and Geometry for Scene Logo Retrieval," in *Computer and Information Science: Proceedings of the 25th International Symposium on Computer and Information Sciences*, vol. 62, p. 305, Springer-Verlag New York Inc, 2010.

[49] M. Soysal and A. A. Alatan, "Joint Utilization of Appearance, Geometry and Chance for Scene Logo Retrieval," *The Computer Journal*, vol. 54, no. 7, pp. 1221–1231, 2011.

[50] M. Soysal and A. Alatan, "Multiview scene matching using local features and invariant geometric constraints," in *Proceedings of the 20th IEEE Signal Processing and Communications Applications Conference (SIU)*, 2012. to appear.

[51] M. Soysal and A. Alatan, "3D Object Recognition using Covariant Local Appearance Descriptors and Invariant Geometric Constraints," in *European Conference on Computer Vision (ECCV)*, 2012. submitted.

[52] M. Soysal and A. Alatan, "Combining Local Appearance and Geometric Invariants for 3D Object Recognition," in *British Machine Vision Conference (BMVC)*, 2012. submitted.

[53] M. Soysal and A. A. Alatan, *Combining MPEG-7 based visual experts for reaching semantics*, vol. 2849. Springer Berlin / Heidelberg, 2003.

[54] E. Esen, O. Önür, M. Soysal, Y. Yasaroglu, S. Tekinalp, and A. A. Alatan, "A MPEG-7 compliant Video Management System: BilVMS," *Digital Media Processing for Multimedia Interactive Services - Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services*, pp. 77–80, 2003.

[55] D. A. Forsyth, "Benchmarks for storage and retrieval in multimedia databases," *Storage and Retrieval for Media Databases*, 2002.

[56] B. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: multimedia content description interface*, vol. 1. John Wiley & Sons Inc, 2002.

[57] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using Expectation-Maximization and its application to image querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1026–1038, 1999.

[58] P. Viola, "Rapid object detection using a boosted cascade of simple features," pp. 511–518, 2001.

[59] T. Tuytelaars and C. Schmid, "Vector Quantizing Feature Space with a Regular Lattice," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, 2007.

[60] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 604 – 610 Vol. 1, 2005.

[61] R. Maree, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 34 – 40 vol. 1, June 2005.

[62] T. Tuytelaars and K. Mikolajczyk, "A survey on local invariant features," *Foundations and Trends in Computer Graphics and Vision,(3)*, pp. 177–280, 2008.

[63] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, p. 50, Manchester, UK, 1988.

[64] R. Deriche and G. Giraudon, "A Computational Approach for Corner and Vertex Detection," *International Journal of Computer Vision*, vol. 10, pp. 101–124, 1992.

[65] S. M. Smith and J. M. Brady, "SUSAN - A New Approach to Low Level Image Processing," *International Journal of Computer Vision*, vol. 23, pp. 45–78, 1995.

[66] D. Marr, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. W.H. Freeman, 1982.

[67] P. Rosin, "Measuring Corner Properties," in *Computer Vision & Image Understanding, Vol.73, No.2*, pp. 291–307, 1999.

[68] E. R. Davies, "Application of the generalised Hough transform to corner detection," *Computers and Digital Techniques, IEE Proceedings E*, vol. 135, pp. 49–54, Jan. 1988.

[69] J. Matas, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, pp. 761–767, Sept. 2004.

[70] T. Tuytelaars and L. Van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *British Machine Vision Conference*, pp. 412–425, Citeseer, 2000.

[71] T. Tuytelaars and K. Mikolajczyk, "Local Invariant Feature Detectors: A Survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2007.

[72] E. Rosten, "Fusing points and lines for high performance tracking," *Computer Vision, 2005. ICCV 2005.*, pp. 1508–1515 Vol. 2, 2005.

[73] E. Rublee, V. Rabaud, and K. Konolige, "ORB: an efficient alternative to SIFT or SURF," *Computer Vision (ICCV*, 2011.

[74] K. Mikolajczyk, "Scale & Affine Invariant Interest Point Detectors," *International Journal of Computer Vision*, vol. 60, pp. 63–86, Oct. 2004.

[75] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.

[76] H. Bay, a. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, June 2008.

[77] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," *Computer Vision-ECCV 2004*, pp. 228–241, 2004.

[78] T. Tuytelaars and L. Van Gool, "Matching Widely Separated Views Based on Affine Invariant Regions," *International Journal of Computer Vision*, vol. 59, pp. 61–85, Aug. 2004.

[79] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local features for object class recognition," *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pp. 1792–1799, 2005.

[80] M. Asbach, P. Hosten, and M. Unger, *An Evaluation of Local Features for Face Detection and Localization*. IEEE, 2008.

[81] A. Vedaldi, "VLFeat.org (last accessed 01.05.2012)." www.vlfeat.org.

[82] A. P. Ashbrook, N. A. Thacker, P. I. Rockett, and C. I. Brown, "Robust recognition of scaled shapes using pairwise geometric histograms," in *Proceedings of the 6th British conference on Machine vision (Vol. 2)*, BMVC '95, (Surrey, UK, UK), pp. 503–512, BMVA Press, 1995.

[83] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 509–522, Apr. 2002.

[84] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, (Washington, DC, USA), pp. 886–893, IEEE Computer Society, 2005.

[85] A. E. Johnson and M. Hebert, "Recognizing objects by matching oriented points," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 684–689, June 1997.

[86] S. Lazebnik, C. Schmid, and J. Ponce, "Affine-invariant local descriptors and neighborhood statistics for texture recognition," *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 649–655 vol.1, 2003.

[87] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 971–987, July 2002.

[88] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recogn.*, vol. 24, pp. 1167–1186, Dec. 1991.

[89] X. Wu and B. Bhanu, "Gabor wavelets for 3-D object recognition," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pp. 537–542, June 1995.

[90] L. M. J. Florack, B. M. Ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "General intensity transformations and differential invariants," *Journal of Mathematical Imaging and Vision*, vol. 4, no. 2, pp. 171–187, 1994.

[91] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, pp. 891–906, Sept. 1991.

[92] A. Baumberg, "Reliable Feature Matching across Widely Separated Views," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, p. 1774, 2000.

[93] L. J. V. Gool, T. Moons, and D. Ungureanu, "Affine/ Photometric Invariants for Planar Intensity Patterns," in *Proceedings of the 4th European Conference on Computer Vision-Volume I - Volume I*, ECCV '96, (London, UK), pp. 642–651, Springer-Verlag, 1996.

[94] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele, "An evaluation of local shape-based features for pedestrian detection," in *In Proc. BMVC*, Citeseer, 2005.

[95] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[96] M. Brown and D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *Int. J. Comput. Vision*, vol. 74, pp. 59–73, Aug. 2007.

[97] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-View Spatial Constraints," in *CVPR (2)*, pp. 272–280, 2003.

[98] P. Pritchett and A. Zisserman, "Wide baseline stereo matching," *Sixth International Conference on Computer Vision*, pp. 754–760, 2004.

[99] L. Fei-Fei, R. Fergus, and A. Torralba, "Short Course: Recognizing and Learning Object Categories," in *ICCV*, (Kyoto Japan), 2009.

[100] M. A. Fischler and R. A. Elschlager, "The Representation and Matching of Pictorial Structures," *IEEE Trans. Comput.*, vol. 22, pp. 67–92, Jan. 1973.

[101] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II–264 – II–271 vol.2, June 2003.

[102] B. Leibe, A. Leonardis, and B. Schiele, "Combined Object Categorization and Segmentation With An Implicit Shape Model," in *In ECCV workshop on statistical learning in computer vision*, pp. 17–32, 2004.

[103] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[104] M. C. Burl, T. K. Leung, and P. Perona, "Face localization via shape statistics," *Constellations*, 1995.

[105] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *In ECCV*, pp. 18–32, 2000.

[106] S. Maji and J. Malik, "Object detection using a max-margin Hough transform," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1038–1045, June 2009.

[107] B. Leibe, A. Ettlin, and B. Schiele, "Learning semantic object parts for object categorization," *Image Vision Comput.*, vol. 26, pp. 15–26, Jan. 2008.

[108] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, Sept. 2009.

[109] "Oxford University Visual Geometry Group (last accessed 01.05.2012)." www.robots.ox.ac.uk.

[110] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22, 2004.

[111] D. D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," pp. 4–15, Springer Verlag, 1998.

[112] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha, "Real-Time Visual Concept Classification," *IEEE Transactions on Multimedia*, vol. 12, pp. 665–681, Nov. 2010.

[113] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Pattern Classification and Scene Analysis: Pattern Classification, Wiley, 2001.

[114] M. Kearns, Y. Mansour, and A. Y. Ng, "An information-theoretic analysis of hard and soft assignment methods for clustering," in *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, (Norwell, MA, USA), pp. 495–520, Kluwer Academic Publishers, 1998.

[115] E. Nowak, F. Jurie, and B. Triggs, "Sampling Strategies for Bag-of-Features Image Classification," *Computer Vision - ECCV 2006*, pp. 490–503, 2006.

[116] J. Zhang, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, p. 2007, 2007.

[117] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek, "Visual Word Ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271–1283, 2010.

[118] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple Kernels for Object Detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[119] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3539–3546, June 2010.

[120] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, (Berlin, Heidelberg), pp. 143–156, Springer-Verlag, 2010.

[121] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[122] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning Object Categories from Google's Image Search.," in *Proceedings of the 10th International Conference on Computer Vision, Beijing, China*, vol. 2, pp. 1816–1823, Oct. 2005.

[123] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1458 –1465 Vol. 2, 2005.

[124] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, (Washington, DC, USA), pp. 2169–2178, IEEE Computer Society, 2006.

[125] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, (New York, NY, USA), pp. 401–408, ACM, 2007.

[126] "ENS/INRIA Visual Recognition and Machine Learning Summer School 2011 (last accessed 01.05.2012)." http://www.di.ens.fr/willow/events/cvml2011.

[127] A. Torralba, K. P. Murphy, and W. T. Freeman, "Using the forest to see the trees: exploiting context for visual object detection and localization," *Commun. ACM*, vol. 53, pp. 107–114, Mar. 2010.

[128] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1365–1372, IEEE, 2009.

[129] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," *Computer Vision - ECCV 2010*, pp. 168–181, 2010.

[130] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, pp. 509–517, Sept. 1975.

[131] J. Beis and D. G. Lowe, "Indexing without invariants in 3D object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1000–1015, 1999.

[132] M. Greenspan and M. Yurick, "Approximate k-d tree search for efficient ICP," in *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pp. 442–448, 2003.

[133] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pp. 331–340, INSTICC Press, 2009.

[134] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala, "Locality-preserving hashing in multidimensional spaces," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, (New York, NY, USA), pp. 618–625, ACM, 1997.

[135] F. Schaffalitzky and A. Zisserman, "Viewpoint Invariant Scene Retrieval using Textured Regions," in *Dagstuhl Seminar on Content-based Image and Video Retrieval* (R. C. Veltkamp, ed.), LNCS, pp. 11–24, Springer-Verlag, 2004.

[136] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous Object Recognition and Segmentation by Image Exploration," in *Toward Category-Level Object Recognition* (J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, eds.), vol. 4170 of *Lecture Notes in Computer Science*, pp. 145–169, Springer Berlin / Heidelberg, 2006.

[137] D. Chen, Y. Chen, and T. Wang, "Moving object detection by multi-view geometric constraints and flow vector classification," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pp. 1630–1634, 2010.

[138] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981.

[139] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, (Berlin, Heidelberg), pp. 500–513, Springer-Verlag, 2008.

[140] O. Chum and J. Matas, "Optimal Randomized RANSAC," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 8, pp. 1472–1482, 2008.

[141] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

[142] H. Wolfson, "Generalizing the generalized hough transform," *Pattern Recognition Letters*, vol. 12, pp. 565–573, Sept. 1991.

[143] H. Jegou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," in *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, (Berlin, Heidelberg), pp. 304–317, Springer-Verlag, 2008.

[144] A. Lehmann, B. Leibe, and L. Van Gool, "Fast PRISM: Branch and Bound Hough Transform for Object Class Detection," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 175–197, 2011.

[145] O. Barinova, V. S. Lempitsky, and P. Kohli, "On detection of multiple object instances using Hough transforms.," in *CVPR'10*, pp. 2233–2240, 2010.

[146] T. Quack, V. Ferrari, and L. J. V. Gool, "Video Mining with Frequent Itemset Configurations.," in *CIVR'06*, pp. 360–369, 2006.

[147] T. Quack, V. Ferrari, B. Leibe, and L. J. V. Gool, "Efficient Mining of Frequent and Distinctive Feature Configurations.," in *ICCV'07*, pp. 1–8, 2007.

[148] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, SIGMOD '93, (New York, NY, USA), pp. 207–216, ACM, 1993.

[149] I. Weiss and M. Ray, "Model-based recognition of 3D objects from one view," *Computer Vision-ECCV98*, vol. 23, no. 2, pp. 716–732, 1998.

[150] W. YuanBin, Z. Bin, and Y. Ge, "The Invariant Relations of 3D to 2D Projection of Point Sets," *Journal of Pattern Recognition Research*, vol. 1, pp. 14–23, 2008.

[151] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall series in artificial intelligence, Prentice Hall, 2003.

[152] A. Joly and O. Buisson, "Logo retrieval with a contrario visual query expansion," in *Proceedings of the 17th ACM international conference on Multimedia*, MM '09, (New York, NY, USA), pp. 581–584, ACM, 2009.

[153] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-local Affine Parts for Object Recognition," in *British Machine Vision Conference (BMVC '04)* (A. Hoppe, S. Barman, and T. Ellis, eds.), (Kingston, United Kingdom), pp. 779–788, The British Machine Vision Association (BMVA), 2004.

[154] S. Lazebnik, C. Schmid, and J. Ponce, "A Maximum Entropy Framework for Part-Based Texture and Object Recognition," in *10th International Conference on Computer Vision (ICCV '05)*, vol. 1, (Beijing, China), pp. 832–838, IEEE Computer Society, 2005.

[155] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool, "Towards multi-view object class detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CPRV '06)*, vol. 2, (New York, United States), pp. 1589–1596, IEEE, 2006.

[156] T. Tuytelaars and L. J. V. Gool, "Content-Based Image Retrieval Based on Local Affinely Invariant Regions," in *Proceedings of the Third International Conference on Visual Information and Information Systems*, VISUAL '99, (London, UK), pp. 493–500, Springer-Verlag, 1999.

[157] J. Mutch and D. G. Lowe, "Object Class Recognition and Localization Using Sparse Features with Limited Receptive Fields," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45–57, 2008.

[158] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," *Int. J. Comput. Vision*, vol. 81, pp. 2–23, Jan. 2009.

[159] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," in *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, (London, UK, UK), pp. 128–142, Springer-Verlag, 2002.

[160] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II – 409–15 vol.2, June 2003.

[161] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results." http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[162] S. Seiden, M. Dillencourt, S. Irani, R. Borrey, and T. Murphy, "Logo Detection in Document Images," in *In: Proceedings of the International Conference on Imaging Science, Systems, and Technology*, pp. 446–449, 1997.

[163] F. A. Albiol, M. J. Ch, and L. Torres, "Detection of TV commercials," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 3, pp. iii – 541–4 vol.3, May 2004.

[164] E. Esen, M. Soysal, T. K. Ates, A. Saracoglu, and A. Aydin Alatan, "A fast method for animated TV logo detection," in *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pp. 236–241, June 2008.

[165] B. Kovar and A. Hanjalic, "Logo detection and classification in a sport video: video indexing for sponsorship revenue control.," in *Storage and Retrieval for Media Databases'02*, pp. 183–193, 2002.

[166] C. M. Bishop, *Pattern recognition and machine learning*. Information science and statistics, Springer, 2006.

[167] C. G. M. Snoek, K. E. A. van de Sande, O. de Rooij, B. Huurnink, J. van Gemert, J. R. R. Uijlings, J. He, X. Li, I. Everts, V. Nedovic, M. van Liempt, R. van Balen, M. de Rijke, J.-M. Geusebroek, T. Gevers, M. Worring, A. W. M. Smeulders, D. Koelma, F. Yan, M. A. Tahir, K. Mikolajczyk, and J. Kittler, "The MediaMill TRECVID 2008 Semantic Video Search Engine.," in *TRECVID'08*, pp. 1–1, 2008.

[168] J. Sivic and A. Zisserman, "Video data mining using configurations of viewpoint invariant regions," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–488 – I–495 Vol.1, 2004.

[169] de Jesus and J. Facon, "Segmentation of Brazilian Bank Check Logos without a Priori Knowledge," in *Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'00)*, ITCC '00, (Washington, DC, USA), pp. 259—-, IEEE Computer Society, 2000.

[170] D. Doermann, E. Rivlin, and I. Weiss, "Logo recognition using geometric invariants," *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, pp. 894–897, 1993.

[171] K. Meisinger, T. Troeger, M. Zeller, and A. Kaup, "Automatic TV logo removal using statistical based logo detection and frequency selective inpainting," in *European Signal Processing Conference (EUSIPCO)*, 2005.

[172] J. Wang, L. Duan, Z. Li, J. Liu, H. Lu, and J. S. Jin, "A Robust Method for TV Logo Tracking in Video Streams," in *Multimedia and Expo, 2006 IEEE International Conference on*, pp. 1041–1044, July 2006.

[173] A. D. Santos and H. Kim, "Real-Time Opaque and Semi-Transparent TV Logos Detection," *I2TS*, 2004.

[174] T. K. Ates, E. Esen, A. Saracoglu, and A. A. Alatan, "Boundary matching based translucent TV logo detection," in *Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th*, pp. 1–4, Apr. 2008.

[175] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J.-M. Morel, "An A Contrario Decision Method for Shape Element Recognition," *Int. J. Comput. Vision*, vol. 69, pp. 295–315, Sept. 2006.

[176] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[177] A. P. Psyllos, C.-N. E. Anagnostopoulos, and E. Kayafas, "Vehicle Logo Recognition Using a SIFT-Based Enhanced Matching Scheme," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 322–328, June 2010.

[178] V. Ferrari, T. Tuytelaars, and L. Gool, "Simultaneous Object Recognition and Segmentation from Single or Multiple Model Views," *International Journal of Computer Vision*, vol. 67, pp. 159–188, Jan. 2006.

[179] S. Hinterstoisser, S. Benhimane, , V. Lepetit, and N. Navab, "Simultaneous Recognition and Homography Extraction of Local Patches with a Simple Linear Classifier," *BMVC British Machine Vision Conference 2008*, 2008.

[180] M. Varma and D. Ray, "Learning The Discriminative Power-Invariance Trade-Off," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, 2007.

[181] S. Hinterstoisser S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal Templates for Real-Time Detection of Texture-less Objects in Heavily Cluttered Scenes," *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[182] G. Strang, *Introduction to linear algebra*. Wellesley, MA USA: Wellesley-Cambridge Press, 2003.

APPENDIX A

DERIVATION OF INVARIANTS AND INVARIANT RELATIONS

In this appendix, geometric invariants are derived for point sets in affine and projective spaces. These derivations are performed in a way to enable formation of relations between invariants of point sets in 3D spaces and their projections in 2D. The derivations included in this chapter reproduces the work of Weiss [149].

A.1 Linear Algebraic Prerequisites

Linear algebraic operations constitute an important part of the derivations in Sections A.2 and A.3. This section is devoted to theorems of linear algebra that are essential for the aforementioned derivations. The first theorem below is related to the *n-linearity* property of determinants on $n \times n$ matrices, and the ones that follow are related to the relations governing the *product of determinants* and *row operations of determinants*.

Theorem A.1.1 *Determinant is an n-linear function on $n \times n$ matrices [182], that is to say, if the i^{th} row of a square matrix is of the form $cR_i + dR'_i$ where R_i and R'_i are $1 \times n$ matrices, then we have:*

$$\begin{vmatrix} R_1 \\ \vdots \\ cR_i + dR'_i \\ \vdots \\ R_n \end{vmatrix} = c \begin{vmatrix} R_1 \\ \vdots \\ R_i \\ \vdots \\ R_n \end{vmatrix} + d \begin{vmatrix} R_1 \\ \vdots \\ R'_i \\ \vdots \\ R_n \end{vmatrix} \tag{A.1}$$

Theorem A.1.2 *Determinant is alternating [182], that is to say, the determinant of any matrix with two identical rows is equal to zero. Further, if a multiple of one row is added to another row the determinant does not change; and if two rows are interchanged the determinant is multiplied by (-1).*

Theorem A.1.3 *The determinant of a product is equal to the product of determinants [182]:*

$$|AB| = |A||B| \tag{A.2}$$

These three algebraic properties of determinants will be utilized for the derivations in Sections A.2 and A.3.

A.2 Invariants and Invariant Relations of Affine 3D and 2D Spaces

In this context, 3D homogeneous world coordinates are denoted by $\vec{X}$, and 2D image coordinates by $\vec{x}$. For the invariants of affine space and their relations, at least five 3D points $\vec{X_i}, i = 1, \ldots, 5$, in general setting (no four of which are on the same plane) are required. These five points can not be independent, since in 3D space any point can be represented as a linear combination of four independent points. In accordance with this fact, $\vec{X_5}$ can be represented uniquely as a linear combination of the others:

$$\vec{X_5} = a\vec{X_1} + b\vec{X_2} + c\vec{X_3} + d\vec{X_4} \tag{A.3}$$

The combination weights a, b, c, d are constrained by the fact that fourth homogeneous coordinate is always 1, which can also be represented mathematically by the constraining relation:

$$a + b + c + d = 1 \tag{A.4}$$

Determinant operation is a linear function in both 3D and 2D, and therefore determinants are relative invariants of transformations in these spaces. The determinant of matrices involving the original and transformed entities are related through a constant scale factor based, which is equal to the determinant of the affine transformation matrix (Theorem A.1.3). Any four of the five points in 3D, represented by homogeneous coordinates can form a determinant. Let us denote determinant of the 4×4 matrix formed by an ordered quadruple of 3D point

coordinates, as M_i. For instance, determinant of the matrix formed by the first four points, $(\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5})$, can be defined as:

$$M_1 = \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5}\right| \tag{A.5}$$

Determinants M_i are indexed using the index of the point that is left out (first point, $\vec{X_1}$ in the above case). According to this convention, M_2, M_3, M_4 and M_5 are defined as:

$$M_2 = \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_5}\right| \tag{A.6}$$

$$M_3 = \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_5}\right| \tag{A.7}$$

$$M_4 = \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_5}\right| \tag{A.8}$$

$$M_5 = \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| \tag{A.9}$$

As indicated by Equation (A.3), there is a dependence among 3D points. Determinants involving these points are also related due to the n-linearity property of determinants (Theorem A.1.1). Substituting the dependence in Equation (A.3), and augmenting the point coordinates with $A = \left[\vec{X_2}, \vec{X_3}, \vec{X_4}\right]$ the determinant $M_1 = \left|A|\vec{X_5}\right|$, can be represented in terms of $\left|A|\vec{X_i}\right|$ as:

$$M_1 = a\left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| + b\left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_2}\right| + c\left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_3}\right| + d\left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_4}\right| \tag{A.10}$$

Due to the property of determinants that is explained in Theorem A.1.2, any determinant with two identical columns vanishes. In addition, according to the same theorem, when columns are interchanged in a determinant, the value of the determinant is multiplied by (-1). Using these, we simplify Equation (A.10) as:

$$M_1 = a\left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| = -a\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| = -aM_5 \tag{A.11}$$

In a manner similar to Equation (A.10),The dependence Equation (A.3) can be substituted in the determinants M_2, M_3, M_4 defined in Equations (A.6) through (A.8) as follows:

$$M_2 = a\left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| + b\left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_2}\right| + c\left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_3}\right| + d\left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_4}\right| \tag{A.12}$$

$$M_3 = a\left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_1}\right| + b\left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_2}\right| + c\left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_3}\right| + d\left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_4}\right| \tag{A.13}$$

$$M_4 = a\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_1}\right| + b\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_2}\right| + c\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_3}\right| + d\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| \tag{A.14}$$

Equations (A.12) through (A.14) can be simplified using Theorem A.1.2 similar to Equation (A.11) as follows:

$$M_2 = b\left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_2}\right| = b\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| = bM_5 \tag{A.15}$$

$$M_3 = c\left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_3}\right| = -c\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| = -cM_5 \tag{A.16}$$

$$M_4 = d\left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| = dM_5 \tag{A.17}$$

Using the results in Equations (A.11), (A.15), (A.16), and (A.17), the weights a, b, c, d in Equation (A.3) can be defined in terms of the ratios of determinants M_i. These coefficients are also invariants of 3D affine transformations.

$$a = -\frac{M_1}{M_5},\ b = \frac{M_2}{M_5},\ c = -\frac{M_3}{M_5},\ d = \frac{M_4}{M_5} \tag{A.18}$$

Projection from 3D to 2D is also a linear operation in homogeneous coordinates (Section 3.2). Due to this linearity, the relation among 3D coordinates (Equation A.3) also hold for the 2D coordinates of their projections:

$$\vec{x_5} = a\vec{x_1} + b\vec{x_2} + c\vec{x_3} + d\vec{x_4} \tag{A.19}$$

Similar to the 3D case, determinants of matrices formed by the 2D homogeneous coordinates, $\vec{x_i}$, of projected points possess the property of relative invariance against linear transformations. Any three of the five point projections can form a determinant, m_{ij}, in which indices i and j correspond to indices of points that are not included. Using this convention, three determinants, m_{12}, m_{13} and m_{14} are defined as:

$$m_{12} = \left|\vec{x_3}, \vec{x_4}, \vec{x_5}\right|, \quad m_{13} = \left|\vec{x_2}, \vec{x_4}, \vec{x_5}\right|, \quad m_{14} = \left|\vec{x_2}, \vec{x_3}, \vec{x_5}\right| \tag{A.20}$$

2D dependence relation in Equation (A.19) can be substituted in the determinants m_{12}, m_{13} and m_{14} by augmenting the point coordinates with a matrix A from the left. For the determinant m_{12}, matrix $A = [\vec{x_3}, \vec{x_4}]$ is selected as the determinant and $m_{12} = \left|A|\vec{x_5}\right|$ can be represented in terms of $\left|A|\vec{x_i}\right|$ as:

$$\begin{aligned} m_{12} &= \left|\vec{x_3}, \vec{x_4}, \vec{x_5}\right| = a\left|\vec{x_3}, \vec{x_4}, \vec{x_1}\right| + b\left|\vec{x_3}, \vec{x_4}, \vec{x_2}\right| + c\left|\vec{x_3}, \vec{x_4}, \vec{x_3}\right| + d\left|\vec{x_3}, \vec{x_4}, \vec{x_4}\right| \\ &= a\left|\vec{x_3}, \vec{x_4}, \vec{x_1}\right| + b\left|\vec{x_3}, \vec{x_4}, \vec{x_2}\right| \\ &= a\left|\vec{x_1}, \vec{x_3}, \vec{x_4}\right| + b\left|\vec{x_2}, \vec{x_3}, \vec{x_4}\right| \\ &= am_{25} + bm_{15} \end{aligned} \tag{A.21}$$

m_{13} and m_{14} are also substituted similarly into Equation (A.19). For $A = [\vec{x_2}, \vec{x_4}]$ and $A = [\vec{x_2}, \vec{x_3}]$ are used for m_{13} and m_{14} respectively. These substitution and simplifications are given in Equation (A.22) and (A.23) below:

$$\begin{aligned} m_{13} &= \left|\vec{x_2}, \vec{x_4}, \vec{x_5}\right| = a\left|\vec{x_2}, \vec{x_4}, \vec{x_1}\right| + b\left|\vec{x_2}, \vec{x_4}, \vec{x_2}\right| + c\left|\vec{x_2}, \vec{x_4}, \vec{x_3}\right| + d\left|\vec{x_2}, \vec{x_4}, \vec{x_4}\right| \\ &= a\left|\vec{x_2}, \vec{x_4}, \vec{x_1}\right| + c\left|\vec{x_2}, \vec{x_4}, \vec{x_3}\right| \\ &= a\left|\vec{x_1}, \vec{x_2}, \vec{x_4}\right| - c\left|\vec{x_2}, \vec{x_3}, \vec{x_4}\right| \\ &= am_{35} - cm_{15} \end{aligned} \tag{A.22}$$

$$\begin{aligned} m_{14} &= \left|\vec{x_2}, \vec{x_3}, \vec{x_5}\right| = a\left|\vec{x_2}, \vec{x_3}, \vec{x_1}\right| + b\left|\vec{x_2}, \vec{x_3}, \vec{x_2}\right| + c\left|\vec{x_2}, \vec{x_3}, \vec{x_3}\right| + d\left|\vec{x_2}, \vec{x_3}, \vec{x_4}\right| \\ &= a\left|\vec{x_2}, \vec{x_3}, \vec{x_1}\right| + d\left|\vec{x_2}, \vec{x_3}, \vec{x_4}\right| \\ &= a\left|\vec{x_1}, \vec{x_2}, \vec{x_3}\right| + d\left|\vec{x_2}, \vec{x_3}, \vec{x_4}\right| \\ &= am_{45} + dm_{15} \end{aligned} \tag{A.23}$$

As previously stated, the determinants formed by the 3D and 2D point coordinates, namely M_i and m_{ij} are relative invariants of affine transformations applied to the points in these spaces. In other words, a 3D affine transformation will merely multiply all the M_i by the same constant factor, while a 2D affine transformation multiplies all the m_{ij} with another constant factor. In order to obtain invariants in 3D and 2D affine spaces by dropping out the constant factors, it is required to use the ratios of determinants. In addition, the relations or dependencies among 3D invariants and those among 2D invariants can be linked together via coefficients a, b, c, d. In order to constitute this relation in terms of the ratios of determinants, the coefficients in Equation (A.18) is substituted into Equations (A.21) and (A.22). As a result of this substitution operation, the following relations between 3D and 2D determinants are obtained:

$$m_{12} + \frac{M_1}{M_5}m_{25} - \frac{M_2}{M_5}m_{15} = 0 \tag{A.24}$$

$$m_{13} + \frac{M_1}{M_5}m_{35} - \frac{M_3}{M_5}m_{15} = 0 \tag{A.25}$$

Equation (A.23) is linearly dependent on the other two, due to the constraint given in Equation (A.4), which shows that the coefficients a, b, c, d sum to one. We can divide Equations (A.24) and (A.25) by m_{15} in prder to group the determinants as ratios:

$$\frac{m_{12}}{m_{15}} + \frac{M_1}{M_5}\frac{m_{25}}{m_{15}} - \frac{M_2}{M_5} = 0 \tag{A.26}$$

$$\frac{m_{13}}{m_{15}} + \frac{M_1}{M_5}\frac{m_{35}}{m_{15}} - \frac{M_3}{M_5} = 0 \tag{A.27}$$

The ratios of M_i and m_{ij} determinants in the above equations are invariants of 3D or 2D transformations respectively. For notational convenience, 3D invariants are defined as:

$$I_1 = \frac{M_1}{M_5}, \quad I_2 = \frac{M_2}{M_5}, \quad I_3 = \frac{M_3}{M_5} \tag{A.28}$$

while 2D invariants are defined as:

$$i_1 = \frac{m_{12}}{m_{15}}, \quad i_2 = \frac{m_{13}}{m_{15}}, \quad i_3 = \frac{m_{25}}{m_{15}}, \quad i_4 = \frac{m_{35}}{m_{15}} \tag{A.29}$$

Using these convenient definitions of 3D and 2D invariants, the relations between them (Equation A.26 and A.27) for a set of five 3D points and their affine projections can be rewritten as:

$$i_1 + I_1 i_3 - I_2 = 0 \tag{A.30}$$

$$i_2 + I_1 i_4 - I_3 = 0 \tag{A.31}$$

A.3 Invariants and Invariant Relations of Projective 3D and 2D Spaces

Geometric invariants of 3D and 2D projective space and relation linking them are derived similar to the affine case. However, due to the higher degree of freedom existent in projective transformations, six 3D points and their projections in 2D image coordinates are necessary for the relation among them. 3D homogeneous world coordinates are denoted by $\vec{X}$, and 2D image coordinates by $\vec{x}$ like the affine case. In projective 3D space, two points $\vec{X_5}$ and $\vec{X_6}$ in 3D space can be represented as a combination of four independent points $\vec{X_1}$, $\vec{X_2}$, $\vec{X_3}$ and $\vec{X_4}$ with a higher number of coefficients:

$$\lambda_5 \vec{X_5} = a\lambda_1 \vec{X_1} + b\lambda_2 \vec{X_2} + c\lambda_3 \vec{X_3} + d\lambda_4 \vec{X_4} \tag{A.32}$$

$$\lambda_6 \vec{X_6} = a'\lambda_1 \vec{X_1} + b'\lambda_2 \vec{X_2} + c'\lambda_3 \vec{X_3} + d'\lambda_4 \vec{X_4} \tag{A.33}$$

As indicated in Equations (A.32) and (A.33), points in projective 3-space, which are represented by homogeneous coordinates can not be uniquely defined by Equation (A.3). Instead, the coordinates (a, b, c, d) can be multiplied by an arbitrary non-zero factor, still satisfying the equation. Therefore, this factor need to be eliminated in order to obtain projective 3D invariants.

In order to obtain 3D invariants, at least six points are needed according to the number of invariants calculation ($3\times6-15=3$). Invariants of six points can be derived using determinant-based calculations similar to the affine case. For this, nine determinant definitions are utilized. Similar to the affine case five of these determinants M_i are indexed using the index of the point that is left out (first point, $\vec{X_1}$ in the above case). According to this convention, M_1, M_2, M_3, M_4 and M_5 are defined as:

$$\begin{aligned} M_1 &= \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_5}\right| \\ M_2 &= \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_5}\right| \\ M_3 &= \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_5}\right| \\ M_4 &= \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_5}\right| \\ M_5 &= \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| \end{aligned} \tag{A.34}$$

In addition to these five determinants, four more are defined involving the sixth point $\vec{X_6}$, instead of the fifth point $\vec{X_5}$:

$$\begin{aligned} M'_1 &= \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_6}\right| \\ M'_2 &= \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_6}\right| \\ M'_3 &= \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_6}\right| \\ M'_4 &= \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_6}\right| \end{aligned} \tag{A.35}$$

As indicated by Equation (A.32), there is a dependence among 3D points. Determinants involving these points are also related due to the n-linearity property of determinants (Theorem A.1.1). Substituting the dependence in Equation (A.32), and augmenting the point coordinates with $A = \left[\vec{X_2}, \vec{X_3}, \vec{X_4}\right]$ the determinant $M_1 = \left|A|\vec{X_5}\right|$, can be represented in terms of $\left|A|\vec{X_i}\right|$ as:

$$\begin{aligned} \lambda_5 M_1 = a\lambda_1 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| + b\lambda_2 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_2}\right| \\ +c\lambda_3 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_3}\right| + d\lambda_4 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_4}\right| \end{aligned} \tag{A.36}$$

Due to the property of determinants that is explained in Theorem A.1.2, any determinant with two identical columns vanishes. In addition, according to the same theorem, when columns are interchanged in a determinant, the value of the determinant is multiplied by (-1). Using

these, we simplify Equation (A.10) as:

$$\begin{aligned}\lambda_5 M_1 &= a\lambda_1 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| \\ &= -a\lambda_1 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right| \\ &= -a\lambda_1 M_5\end{aligned} \tag{A.37}$$

The dependence Equations (A.32) and (A.33) are substituted in determinants M_2, M_3, M_4 and M_1', M_2', M_3', M_4' respectively in a manner similar to Equation (A.36):

$$\begin{aligned}\lambda_5 M_2 = a\lambda_1 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| + b\lambda_2 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_2}\right| \\ +c\lambda_3 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_3}\right| + d\lambda_4 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_4}\right|\end{aligned} \tag{A.38}$$

$$\begin{aligned}\lambda_5 M_3 = a\lambda_1 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_1}\right| + b\lambda_2 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_2}\right| \\ +c\lambda_3 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_3}\right| + d\lambda_4 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_4}\right|\end{aligned} \tag{A.39}$$

$$\begin{aligned}\lambda_5 M_4 = a\lambda_1 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_1}\right| + b\lambda_2 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_2}\right| \\ +c\lambda_3 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_3}\right| + d\lambda_4 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right|\end{aligned} \tag{A.40}$$

$$\begin{aligned}\lambda_6 M_1' = a'\lambda_1 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| + b'\lambda_2 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_2}\right| \\ +c'\lambda_3 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_3}\right| + d'\lambda_4 \left|\vec{X_2}, \vec{X_3}, \vec{X_4}, \vec{X_4}\right|\end{aligned} \tag{A.41}$$

$$\begin{aligned}\lambda_6 M_2' = a'\lambda_1 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_1}\right| + b'\lambda_2 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_2}\right| \\ +c'\lambda_3 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_3}\right| + d'\lambda_4 \left|\vec{X_1}, \vec{X_3}, \vec{X_4}, \vec{X_4}\right|\end{aligned} \tag{A.42}$$

$$\begin{aligned}\lambda_6 M_3' = a'\lambda_1 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_1}\right| + b'\lambda_2 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_2}\right| \\ +c'\lambda_3 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_3}\right| + d'\lambda_4 \left|\vec{X_1}, \vec{X_2}, \vec{X_4}, \vec{X_4}\right|\end{aligned} \tag{A.43}$$

$$\begin{aligned}\lambda_6 M_4' = a'\lambda_1 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_1}\right| + b'\lambda_2 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_2}\right| \\ +c'\lambda_3 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_3}\right| + d'\lambda_4 \left|\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\right|\end{aligned} \tag{A.44}$$

(A.45)

Simplified versions of Equations (A.36) and (A.38) through (A.44) using Theorem A.1.2 are as follows:

$$\begin{aligned}a\frac{\lambda_1}{\lambda_5} = -\frac{M_1}{M_5}, \quad b\frac{\lambda_2}{\lambda_5} = \frac{M_2}{M_5}, \quad c\frac{\lambda_3}{\lambda_5} = -\frac{M_3}{M_5}, \quad d\frac{\lambda_4}{\lambda_5} = \frac{M_4}{M_5} \\ a'\frac{\lambda_1}{\lambda_6} = -\frac{M_1'}{M_5}, \quad b'\frac{\lambda_2}{\lambda_6} = \frac{M_2'}{M_5}, \quad c'\frac{\lambda_3}{\lambda_6} = -\frac{M_3'}{M_5}, \quad d'\frac{\lambda_4}{\lambda_6} = \frac{M_4'}{M_5}\end{aligned} \tag{A.46}$$

As mentioned above, in projective case invariants are more complex in comparison to the affine case. This is due to the fact that the determinant ratios given above in Equation (A.46)

are not invariant. In order to obtain 3D invariants, cross ratios of determinants are utilized for eliminating all the λ_i terms:

$$I_1 = \frac{ab'}{a'b} = \frac{M_1 M_2'}{M_1' M_2}, \quad I_2 = \frac{ac'}{a'c} = \frac{M_1 M_3'}{M_1' M_3}, \quad I_3 = \frac{ad'}{a'd} = \frac{M_1 M_4'}{M_1' M_4} \tag{A.47}$$

Since projection from 3D to 2D is a linear operation in homogeneous coordinates (Section 3.2), the relations among 3D coordinates (Equations A.32 and A.32) also hold for the 2D coordinates of their projections:

$$\lambda_5 \vec{x_5} = a\lambda_1 \vec{x_1} + b\lambda_2 \vec{x_2} + c\lambda_3 \vec{x_3} + d\lambda_4 \vec{x_4} \tag{A.48}$$

$$\lambda_6 \vec{x_6} = a'\lambda_1 \vec{x_1} + b'\lambda_2 \vec{x_2} + c'\lambda_3 \vec{x_3} + d'\lambda_4 \vec{x_4} \tag{A.49}$$

Similar to the 3D case, in 2D, determinants of matrices formed by any three of the first five 2D homogeneous coordinates, $\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{x_4}, \vec{x_5}$, of projected points can form a determinant, m_{ij}, in which indices i and j correspond to indices of points that are not included. Using a similar convention, but replacing $\vec{x_5}$ with $\vec{x_6}$, we define determinants m'_{ij}. According to these conventions, the definitions of m_{12} and m'_{12} are provided below as examples:

$$m_{12} = \left|\vec{x_3}, \vec{x_4}, \vec{x_5}\right|$$

$$m'_{12} = \left|\vec{x_3}, \vec{x_4}, \vec{x_6}\right|$$

From Theorem A.1.3, we know that determinants are relative invariants of linear transformations, and therefore we know that cross ratios of determinants lead to cancellation of transform related scaling terms, which in turn leads to 2D projective invariants. For convenience in the following steps, we form the following four 2D invariants. We can derive invariants from other determinants, but they will be dependent on the invariants below:

$$i_1 = \frac{m'_{12} m_{14}}{m_{12} m'_{14}}, \quad i_2 = \frac{m'_{12} m_{35}}{m_{25} m'_{13}}, \quad i_3 = \frac{m'_{12} m_{13}}{m_{12} m'_{13}}, \quad i_4 = \frac{m'_{12} m_{45}}{m_{25} m'_{14}} \tag{A.50}$$

Now that we have derived 3D and 2D invariants for the projective transformation case, we can relate them using common terms in their definitions. 3D invariants that are defined in Equation (A.47) are defined in terms of a, b, c, d and a', b', c', d'. In order to relate 2D invariants defined in Equation (A.50), we need to use the method of substituting dependency relations (Equation A.48 and A.49) into the definitions of determinants m_{12}, m_{13}, m_{14} and $m'_{12}, m'_{13}, m'_{14}$, like we did in Equations (A.21), (A.22) and (A.23).

Like the affine case, 2D dependence relations in Equations (A.48) and (A.49) can be substituted in the determinants m_{12}, m_{13}, m_{14} and $m'_{12}, m'_{13}, m'_{14}$ by augmenting the point coordinates with a matrix A from the left. For the determinant m_{12}, matrix $A = [\vec{x_3}, \vec{x_4}]$ is selected as the determinant and $m_{12} = \left|A|\vec{x_5}\right|$ can be represented in terms of $\left|A|\vec{x_i}\right|$ as:

$$\begin{aligned}
\lambda_5 m_{12} &= \lambda_5 \left|\vec{x_3}, \vec{x_4}, \vec{x_5}\right| \\
&= a\lambda_1 \left|\vec{x_3}, \vec{x_4}, \vec{x_1}\right| + b\lambda_2 \left|\vec{x_3}, \vec{x_4}, \vec{x_2}\right| + c\lambda_3 \left|\vec{x_3}, \vec{x_4}, \vec{x_3}\right| + d\lambda_4 \left|\vec{x_3}, \vec{x_4}, \vec{x_4}\right| \\
&= a\lambda_1 \left|\vec{x_3}, \vec{x_4}, \vec{x_1}\right| + b\lambda_2 \left|\vec{x_3}, \vec{x_4}, \vec{x_2}\right| \\
&= a\lambda_1 \left|\vec{x_1}, \vec{x_3}, \vec{x_4}\right| + b\lambda_2 \left|\vec{x_2}, \vec{x_3}, \vec{x_4}\right| \\
&= a\lambda_1 m_{25} + b\lambda_2 m_{15}
\end{aligned} \tag{A.51}$$

Only three of the derived dependence relations can be independent for each of the Equations (A.48) and (A.49). In this derivation, we select m_{12}, m_{13}, m_{14} for Equation (A.48) and $m'_{12}, m'_{13}, m'_{14}$ for Equation (A.49). We have four unknown quantities in each of the three equations sets, namely, $a\frac{\lambda_1}{\lambda_5}, b\frac{\lambda_2}{\lambda_5}, c\frac{\lambda_3}{\lambda_5}, d\frac{\lambda_4}{\lambda_5}$ for the first set and $a'\frac{\lambda_1}{\lambda_6}, b'\frac{\lambda_2}{\lambda_6}, c'\frac{\lambda_3}{\lambda_6}, d'\frac{\lambda_4}{\lambda_6}$ for the second set. This means that for each set of equations, there is one free parameter. For our derivation, we select the following two quantities as the free parameters:

$$\mu = a\frac{\lambda_1}{\lambda_5} \tag{A.52}$$

$$\mu' = a'\frac{\lambda_1}{\lambda_6} \tag{A.53}$$

Substituting the first free parameter, μ, into equations for m_{12} (Equation A.51), m_{13} and m_{14}, we obtain the following relations:

$$b\frac{\lambda_2}{\lambda_5} = \frac{m_{12} - \mu m_{25}}{m_{15}} \tag{A.54}$$

$$c\frac{\lambda_3}{\lambda_5} = \frac{m_{13} - \mu m_{35}}{m_{15}} \tag{A.55}$$

$$d\frac{\lambda_4}{\lambda_5} = \frac{m_{14} - \mu m_{45}}{m_{15}} \tag{A.56}$$

Substituting the second free parameter, μ' into equations for m'_{12}, m'_{13} and m'_{14}, we obtain the following relations:

$$b'\frac{\lambda_2}{\lambda_5} = \frac{m'_{12} - \mu' m_{25}}{m_{15}}$$

$$c'\frac{\lambda_3}{\lambda_5} = \frac{m'_{13} - \mu' m_{35}}{m_{15}}$$

$$d'\frac{\lambda_4}{\lambda_5} = \frac{m'_{14} - \mu' m_{45}}{m_{15}}$$

Using these relations we relate 3D invariants and 2D invariants through three equations:

$$I_1 = \frac{ab'}{a'b} = \frac{\mu\left(m'_{12} - \mu' m_{25}\right)}{\mu'\left(m_{12} - \mu m_{25}\right)}$$
$$I_2 = \frac{ac'}{a'c} = \frac{\mu\left(m'_{13} - \mu' m_{35}\right)}{\mu'\left(m_{13} - \mu m_{35}\right)}$$
$$I_3 = \frac{ad'}{a'd} = \frac{\mu\left(m'_{14} - \mu' m_{45}\right)}{\mu'\left(m_{14} - \mu m_{45}\right)}$$

The free parameters μ and μ' in the above equations can be represented in terms of determinants and I_i in Equations (A.57) and (A.57). These representations are then substituted into Equation (A.57). After this substitution, the determinants in the resulting relation are grouped in order to form cross ratios corresponding to 2D projective invariants in Equation (A.50). The resulting equation below constitutes an invariant relation between 3D invariants that are computed from 3D point coordinates, and 2D invariants that are computed from their projections:

$$I_3(I_2 - 1)i_1 i_2 - I_3(I_1 - 1)i_1 - I_1(I_2 - 1)i_2 = I_2(I_3 - 1)i_3 i_4 - I_2(I_1 - 1)i_3 - I_1(I_3 - 1)i_4 \quad \text{(A.57)}$$

This equation remains invariant under projective or in other words, perspective transformations.

VITA

Medeni Soysal was born in Birmingham, UK in 1980. He received his B.Sc. and M.Sc degrees from Middle East Technical University (METU), Department of Electrical and Electronics Engineering in 2001 and 2003 respectively. He is working towards Ph. D. degree in the same department since 2003.

He joined TÜBİTAK BİLTEN (The Scientific and Technological Research Council of Turkey - Information Technologies Research Institute) in 2002, and worked as a researcher till 2006. Currently, he is working as a chief researcher in Video and Audio Processing Group of TÜBİTAK UZAY (The Scientific and Technological Research Council of Turkey - Space Technologies Research Institute). His areas of interest are computer vision and pattern recognition, more specifically, object recognition, local features, geometric invariants and joint utilization of multi-modal information.

PROFESSIONAL EXPERIENCE

Projects in TÜBİTAK UZAY

- Project Manager in RTÜK SKAAS Semantic Video Classification System Project (SKAAS KAVTAN)
- Comparison Software Module Administrator in Ballistic Image Analysis and Recognition System for Forensic Laboratories Project (Balistika 2010)
- Dimension Reduction and Low-level Description Fusion Module Administrator in RTÜK SKAAS Semantic Video Classification System Project (SKAAS KAVTAN)
- Video Program Analysis Module Administrator in Digital Recording, Archiving and Analysis System Project (RTÜK SKAAS)

Projects in TÜBİTAK BİLTEN

- Project Manager in Automated TV Advertisement Tracking System Project (GÖRETAS)
- Project Manager in Seaborne Platform Recognition from Satellite Images Project (DÜPT)

- Senior Researcher in MPEG-7 Compliant Multimedia Management System Project (BilMMS)
- Researcher in MPEG-7 Compliant Image and Video Management System Project (BilVMS)

PUBLICATIONS

International Peer-Reviewed Journal Papers

- **Soysal, M.** and Alatan, A.A., "Joint Utilization of Appearance, Geometry and Chance for Scene Logo Retrieval," *The Computer Journal*, vol. 54, no. 7, pp. 1221-1231, 2011.

International Conference Papers

- **Soysal, M.**, Alatan, A. A., "Combining Local Appearance and Geometric Invariants for 3D Object Recognition," *submitted to British Machine Vision Conference (BMVC), 2012*.
- **Soysal, M.**, Alatan, A. A., "3D Object Recognition using Covariant Local Appearance Descriptors and Invariant Geometric Constraints," *submitted to European Conference on Computer Vision (ECCV), 2012*.
- **Soysal, M.**, Alatan, A. A., "Joint Utilization of Appearance and Geometry for Scene Logo Retrieval," in International Symposium on Computer and Information Sciences (ISCIS), 22-24 September 2010, London, UK.
- **Soysal, M.**, Alatan, A. A., Karadeniz, T. "Joint Utilization of Appearance and Geometry for Determining Correspondences," in International Symposium on Computer and Information Sciences (ISCIS), 14-16 September 2009, Güzelyurt, Northern Cyprus.
- Esen, E., **Soysal, M.**, Ates, T.K., Saracoglu, A. and Alatan, A.A. "A Fast Method For Animated TV Logo Detection," in Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI), London, UK, 18-20 June 2008, Los Alamitos, CA.
- Örten, B. B., **Soysal, M.**, Alatan, A. A., "Person Identification in Surveillance Video by Combining MPEG-7 Experts," Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), 13-15 April 2005, Montreux, Switzerland.
- **Soysal, M.**, Alatan, A. A., "Combining Region-based MPEG-7 Experts for Reaching Semantics," in COST 276 Workshop (Information and Knowledge Management For Integrated Media Communication), 6-7 May 2004, Thessaloniki, Greece.
- **Soysal, M.**, Alatan, A. A., "Combining MPEG-7 Based Visual Experts For Reaching Semantics," *Lecture Notes in Computer Science*, Proceedings of International Workshop on Very Low Bitrate Video Coding(VLBV), 18-19 September 2003, Madrid, Spain, Vol.87, pp. 66-75.

- Esen, E., Önür, Ö., **Soysal, M.**, Yaşaroğlu, Y., Tekinalp, S. ve Alatan, A.A., "A MPEG-7 Compliant Video Management System," European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), 9-11 April 2003, Queen Mary, University of London, London, UK.

National Conference Papers

- **Soysal, M.**, Alatan, A. A., "Multiview scene matching using local features and invariant geometric constraints," *to appear in Proceedings of the 20th IEEE Signal Processing and Communications Applications Conference (SIU), 18-20 April 2012, Fethiye, Turkey*.
- Saracoglu, A., Tekin, M., Esen, E., **Soysal, M.**, et al., "Multimodal Concept Detection on Multimedia Data - RTÜK SKAAS KavTan System," *to appear in Proceedings of the 20th IEEE Signal Processing and Communications Applications Conference (SIU), 18-20 April 2012, Fethiye, Turkey*.
- Ates, T. K., Özkan, S., **Soysal, M.**, Alatan, A. A., "Relevance Feedback for Semantic Classification: A Comparative Study," in Proceedings of the 19th IEEE Signal Processing and Communications Applications Conference (SIU), 20-22 April 2011, Antalya, Turkey.
- Saracoglu, A., Tekin, M., Esen, E., **Soysal, M.**, et al., "Generalized Concept Detection," in Proceedings of the 18th IEEE Signal Processing and Communications Applications Conference (SIU), 22-24 April 2010, Diyarbakır, Turkey.
- Ates, T. K., Esen, E., Saracoglu , A., **Soysal, M.**, Turgut, Y., Oktay, O., Alatan, A. A., "Content Based Video Copy Detection with Local Descriptors," in Proceedings of the 18th IEEE Signal Processing and Communications Applications Conference (SIU), 22-24 April 2010, Diyarbakır, Turkey.
- Esen, E., Yaşaroğlu, Y., Önür, Ö., **Soysal, M.**, Tekinalp, S., Alatan, A. A., "MPEG-7 Compliant Video Management System," in Proceedings of the 11th IEEE Signal Processing and Communications Applications Conference (SIU), 18-20 June 2003, İstanbul, Turkey.