Correspondence matching in urban scenes across different views

Lee, Jimmy Addison

2011

Lee, J. A. (2011). Correspondence matching in urban scenes across different views. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/44570>

<https://doi.org/10.32657/10356/44570>

NANYANG TECHNOLOGICAL UNIVERSITY

Correspondence Matching in Urban Scenes across Different Views

A Thesis Submitted to the
School of Computer Engineering
of the Nanyang Technological University

by

Jimmy Addison Lee

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy (Ph.D)

2011

Abstract

The high degree of symmetry and repetition in urban environments makes it difficult for computer vision to establish correspondences during wide-baseline matching. The level of difficulty increases with respect to the magnitude of viewpoint changes. This thesis highlights these problems and looks into different approaches to tackle the problems.

First, verification approaches are considered for enhancing the correspondence matching results by eliminating false matches. They comprise geometric and appearance-based measurements, e.g., the Euclidean distance and angle approximations, gabor texture, pattern approximation, etc. to determine the reliability of each match.

Second, we investigate new ways to increase the number of point correspondences regardless of the viewing variations. We define hypotheses of building facades by planar convex quadrilaterals in images, and we call them “q-regions”. A projective transformation model can be derived from each pair of q-regions in two images. If a pair of q-regions is correctly aligned, all line segments and interest points within the pair of q-regions will fit the projective transformation model and match accordingly. Consequently, the largest pair of correctly aligned q-regions will produce the largest list of matches.

Third and last, using a similar concept, we introduce a fully affine invariant descriptor, coined PRIUS (Projective Region Invariants for Urban Scenes). The interest points in the q-regions and their neighboring points within close spatial proximity are used to describe the q-regions, and to robustly align building facades through their projectivity. It has been shown that robustness to false corresponding q-regions and noise is improved, and the number of point correspondences has also increased. The descriptor is substantiated by extensive experiments which demonstrate that it achieves about twenty-fold more true matches than state-of-the-art approaches such as SIFT and Affine-SIFT. We demonstrate the applicability of PRIUS to a proposed mobile guide application.

Acknowledgments

I would like to thank, first and foremost, my Ph.D supervisor Prof. Kin-Choong Yow for his guidance and support throughout my graduate program, to the completion of this thesis. His expertise, understanding, and patience, added considerably to my graduate experience. He becomes more of a mentor and friend than a supervisor. I must also acknowledge other researchers worldwide - Adrian David Cheok, Andrea Vedaldi, Andrzej Sluzek, David Lowe, Itsuo Kumazawa, Ivor Wai-Hung Tsang, Michael Scott Brown, Timo Rolf Bretschneider, and Wei Zhang - for taking time out from their busy schedules to give me insightful suggestions and comments on my work.

A very special thank you goes to my MSc research supervisor Dr. Jane Brennan, without whose motivation and encouragement I would not have considered taking on a Ph.D path. She is the one supervisor who truly made a difference in my life. It was under her tutelage that I developed a keen interest in research. I doubt that I will ever be able to convey my appreciation fully, but I owe her my eternal gratitude.

Next, I would like to thank all the jury members for their time and energy spent on my Ph.D defense. I feel deeply honored by the attention brought to my modest contributions by these experts, and would like to thank them for the interesting comments, questions and discussions before, during, and even after the defense.

Thanks also to my family and friends for their unwavering moral support and endless love. I am particularly indebted to a few awesome friends and lab colleagues - Adeline Chan, Bryan Lee, Chong-Yan Eng, David Goh, Desmond Tsoi, Irene Quek, Poon-Wei Koot, Sing-Heng Teo, Tai-Yu Wing, and Yi Huang - for bearing with me all these years and helping to maintain my sanity throughout my Ph.D journey.

Last but not least, I recognize that this research would not have been possible without the financial assistance, in the form of stipends from the School of Computer Engineering, Nanyang Technological University (Singapore).

Contents

Abstract	i
Acknowledgments	ii
List of Figures	vii
List of Tables	x
1 Introduction	1
1.1 Motivation	1
1.1.1 Symmetry and Repetition	2
1.1.2 Deformation and Occlusion	3
1.2 Approaches in the Thesis	5
1.2.1 Enhancing Image Matches through Verification	5
1.2.2 Invariant Matching across Different Views	5
1.3 Organization of the Thesis	6
2 Literature Review	8
2.1 Introduction	8
2.2 Feature Extraction	9
2.2.1 Moravec detector	10
2.2.2 Harris/Plessey detector	12
2.2.3 Harris-Laplacian detector	15
2.2.4 Harris-Affine detector	17
2.2.5 Difference of Gaussians (DoG)	19
2.2.6 Maximally Stable Extremal Regions (MSER)	24
2.2.7 FAST and FAST-ER detectors	26
2.3 Feature Description	30

2.3.1	Scale Invariant Feature Transform (SIFT)	31
2.3.2	Principle Components Analysis-SIFT (PCA-SIFT)	34
2.3.3	Shape context	36
2.3.4	Gradient Location and Orientation Histogram (GLOH)	38
2.3.5	Speeded Up Robust Features (SURF)	38
2.3.6	Affine-SIFT (ASIFT)	42
2.3.7	Simple grayscale patches	44
2.4	Feature Matching	45
2.4.1	Distance measures	46
2.4.1.1	Minkowski distance	47
2.4.1.2	Histogram intersection	47
2.4.1.3	Chi-square statistic	48
2.4.1.4	Quadratic distance	48
2.4.1.5	Match distance	49
2.4.2	Template matching	50
2.4.2.1	Cross-correlation	50
2.4.2.2	Normalized Cross-Correlation (NCC)	51
2.4.3	Homography matching	52
2.5	Chapter Summary	54
3	Enhancing Image Matches via a Verification Model	55
3.1	Introduction	55
3.2	Related Work	57
3.3	Verification Model	58
3.3.1	Pattern approximation	60
3.3.2	Gabor texture	62
3.3.3	Euclidean distance approximation	65
3.3.4	Angle approximation	66
3.3.5	Final computation of scores	68
3.4	Experimental Results	70
3.5	Chapter Summary	74

4	Image Matching by Combining Invariant and Corner-Like Features	76
4.1	Introduction	76
4.2	Related Work	77
4.3	Combining Invariant and Corner-Like Features	78
4.3.1	Keypoint extraction and reduction	78
4.3.2	Keypoint matching and verification	80
4.4	Experimental Results	81
4.5	Chapter Summary	86
5	Matching of Building Facades under Large Viewpoint Change	87
5.1	Introduction	87
5.2	Related Work	89
5.3	Methodology	93
5.3.1	Line fitting algorithm	96
5.3.2	Q-region hypothesis extraction	96
5.3.2.1	Initial geometric corner detection	97
5.3.2.2	Q-regions formation	98
5.3.2.3	Other geometric corners of the facade	100
5.3.2.4	Possible combinations of q-regions	100
5.3.3	Q-regions alignment	101
5.4	Experimental Setup	105
5.4.1	Other approaches for comparison	105
5.4.2	Evaluation criteria	107
5.4.3	Data sets	108
5.5	Experimental Results	111
5.5.1	Matching under various transformations	111
5.5.1.1	Scale changes	111
5.5.1.2	Orientation changes	113
5.5.1.3	Viewpoint changes	113
5.5.2	Image retrieval	114
5.6	Chapter Summary	118

6	PRIUS: Projective Region Invariants for Urban Scenes	120
6.1	Introduction	120
6.2	Related Work	122
6.3	Methodology	123
6.3.1	PRIUS descriptor and matching	124
6.4	Experimental Results	129
6.5	Mobile Guide Application	133
6.5.1	System Overview	134
6.5.2	Database formulation	135
6.5.3	Pose recovery	135
6.5.4	System Evaluation	136
6.6	Chapter Summary	138
7	Conclusions	140
7.1	Summary and Discussion of Results	140
7.2	Future Work	143
	References	146
	List of Publications	164

List of Figures

1.1	Multiple identical and repeated patterns in urban environments.	3
1.2	Deformation of image patches in an urban scene with respect to viewpoint changes.	4
2.1	Image results from the Moravec detector.	11
2.2	Image results from the Harris detector.	14
2.3	Image results from the Harris-Laplacian detector where points are detected on different resolution levels.	16
2.4	Image results from the Harris-Affine detector.	18
2.5	Scale-space in Difference of Gaussian (DoG).	21
2.6	Detecting Maxima and minima in DoG space.	21
2.7	Image results from the DoG detector, showing stages of keypoint selection.	23
2.8	Image results from the MSER detector where ellipses are fitted to the detected regions.	26
2.9	Corner hypothesis in an image patch by the FAST detector.	27
2.10	Image results from the FAST detector.	29
2.11	Computation of SIFT descriptor.	32
2.12	Image result of location recognition using SIFT descriptor.	34
2.13	Log-polar histogram bins used in computing shape contexts and compact representation of point distribution relative to each bin.	37
2.14	Shape contexts of three edge points.	37
2.15	Haar wavelet types and square regions used for SURF descriptor.	39
2.16	The formulation of the SURF descriptor.	40
2.17	Properties of the SURF descriptor for three distinctively different image intensity patterns within a sub-region.	41

2.18	Overview of the ASIFT algorithm.	42
2.19	Homography mapping of a point on one plane to another plane.	52
3.1	Overall flow diagram of our approach (SIFT + Verification model). . . .	59
3.2	Model configurations of a corner and a T-junction.	60
3.3	Gabor filtered images with random scales and orientations.	64
3.4	A simple illustration of angle change within an Euclidean distance constraint.	66
3.5	Two cases of different angle increment and decrement rates when rotating about the Y-axis.	67
3.6	A simple illustration of angle change within the angle constraints.	68
3.7	Matching results of difficult scenes of four different approaches.	72
3.8	Other matching results of urban scenes of four different approaches. . . .	73
4.1	SIFT samples comprising corner-like and non corner-like patterns.	79
4.2	Complex image pattern responses of two samples.	80
4.3	Two sample groups of small and large complex pattern response differences.	81
4.4	Reduction of keypoints that do not comprise any corner-like patterns. . .	82
4.5	Graph shows matching results from 50 image pairs using SIFT keypoints and predefined keypoints.	83
4.6	Before and after image results of eliminating correspondence pairs with dissimilar complex pattern responses.	84
4.7	Graph shows matching results from 50 image pairs using SIFT keypoints, predefined keypoints, and after further optimization.	85
4.8	2nd data set: Graph shows matching results from 50 image pairs using SIFT keypoints, predefined keypoints, and after further optimization. . .	85
5.1	Hypothesizing a quadrilateral of a building facade.	94
5.2	A quadrilateral is perceived from other corners and edges within the build- ing facade.	95
5.3	Illustration of q-region formation.	97
5.4	Other atypical cases of q-region formation.	99
5.5	Finding q-region with a different initial geometric corner on another side of facade.	101

5.6	All possible combinations of q-regions formed from 4 vertical line segments, giving 6 q-regions.	102
5.7	A projective transformation (homography) model is instantiated by four corresponding corner points in a pair of q-regions, and a point from one q-region is projected to another q-region.	103
5.8	Corresponding line segments and Harris points within a pair of correctly aligned q-regions.	105
5.9	An image example from each of the 9 sets of images used for the evaluation of four approaches on matching under various transformations.	109
5.10	Data set used for the evaluation of four approaches on matching under various transformations.	110
5.11	Evaluation results of four approaches for scale changes of a factor 1.4-2.5.	112
5.12	Evaluation results of four approaches for orientation changes of 10-45 degrees.	112
5.13	Evaluation results of four approaches for viewpoint changes of 15 to > 50 degrees.	114
5.14	Matching results of four approaches on images of building facades with viewpoint change > 50°.	116
5.15	Matching results of four approaches for the images from Table 5.2.	117
6.1	Point projectivity between two quadrilateral regions of the same scene but different camera angles.	126
6.2	Image matching sequence results of four approaches.	130
6.3	Matching results of four approaches for image matching sequence “ae” in Figure 6.2.	131
6.4	Average matching results of the four approaches over 50 images.	132
6.5	Matching results of a pair of false corresponding quadrilateral regions in two images.	133
6.6	The process flow of the mobile guide system.	134
6.7	Examples of location information overlaid onto the query images.	136
6.8	Illustrative of some recovered database views.	137
7.1	Image results of two pairs of corresponding regions detected in a pair of 3D house images from the Visual Geometry Group, Oxford University. .	145

List of Tables

3.1	Experimental values of matched image pairs in percentage (%) of inliers; and the overall average for each of the approaches among the 804 image pair matches.	74
5.1	Percentage of images correctly retrieved in the image retrieval experiment.	115
5.2	An image retrieval example. The top three matches (Rank 1-3) for a query image for the four approaches are shown.	115

Chapter 1

Introduction

1.1 Motivation

Wide-baseline matching in urban scenes is one of the very challenging tasks in computer vision. ***Problems.*** The high degree of symmetry and repetition in these man-made environments introduces a significant number of false matches; while a substantial change in viewpoint induces a deformation of the image scene and hampers the number of matches to a modest number. Albeit studied for decades, we are still far from a solution that is truly feasible on the above problems. ***Challenges.*** Following the Gestalt psychophysical invariance laws proposed by Attneave, Kanizsa, and Wertheimer [8, 62, 166], a good image/correspondence matching algorithm should satisfy the following requirements [25]:

- (i) Robust to partial occlusions,
- (ii) Invariant to illumination changes,
- (iii) Insensitive to the noise inherent to any image acquisition device, and
- (iv) Independent of the viewpoint changes.

A typical correspondence matching process consists of, for each concerned image, first detecting points of interest and selecting a region around each point of interest,

and then associating each region with a feature descriptor. Correspondences may thus be established by matching the feature descriptors [98]. Generally, the feature detectors and descriptors should have some invariance according to the “good matching” requirements above. The first three “good matching” requirements are easier to satisfy. Local descriptors induce robustness to partial occlusion, while features based on directions of gradients or level sets have certain invariance to illumination changes. Noise can be reduced by pre-filtering the images before matching. The fourth requirement (i.e. viewpoint invariance) is however much more challenging, especially in urban scenes where there are additional requirements to fulfill, namely the robustness to symmetry and repetition in the environments [75, 132]. The objective of the thesis is to fulfill all these requirements.

Several studies [56, 68, 75, 102, 111] show that urban scenes often comprise mutual characteristics that represent very important aspects for three-dimensional (3D) inference and image matching. For example, the facades of many structures in the man-made environments consist of some planar quadrilaterals such as buildings, windows, doors, etc. Corners and edges can be extracted from these structures, e.g., the boundaries of scene buildings and windows. Roofs of buildings tend to be parallel to the ground plane, while walls tend to be perpendicular to this plane [56]. Moreover, edges in the scene that are perpendicular to the ground will appear in images to be directed towards the vertical vanishing point. The buildings in the scene also comprise other appearance properties such as color, texture, illumination, etc. A deep understanding of such characteristics is necessary for tackling the problems by perceiving the visual world just as humans do.

1.1.1 Symmetry and Repetition

An obvious observation on the facades of many structures in urban settings is the appearance of symmetric and repeated patterns, e.g., the multiple columns and rows of identical windows in a building as shown in Figure 1.1. The matching between these

patterns from different images is difficult and tedious even for humans since many of the patterns (e.g., windows) look identical. Consequently, false matches are rather inevitable. The existence of false matches is not trivial given that a single false match is enough to generate erroneous results, e.g., in operations such as homographies which are extremely sensitive to noise and false matches [77].

Figure 1.1: Multiple identical and repeated patterns in urban environments. The columns and rows of identical windows yield the appearance of repeated patterns. Three close-up views of identical corner patches are shown. The building image is extracted from the ZuBud database [135].

1.1.2 Deformation and Occlusion

Repeated patterns viewed from different viewpoints, e.g., from the building facades in an urban scene, present computational difficulties when being matched to each other or to a 3D model directly. The primary challenge is the deformation or distortion of the matching patterns. For instance, a team of Penn State researchers [58] state -

“Computer might have reached a point where they can replicate many aspects of human behavior, but still they cannot recognize distorted images like humans do.”

The ambiguity in scale, unit patch, and skewness of the matching patterns hampers the number of matches to a modest number, especially when the viewpoint difference gets larger. Figure 1.2 illustrates the deformation of an image patch of the same scene with respect to the changes in viewpoint.

Figure 1.2: Deformation of image patches in urban scene with respect to viewpoint changes. Three images taken from different viewpoints of the same building are shown. An image patch of the same physical structure is extracted from each image, and the close-up view of each is shown. The deformation in (c) is large due to the significant change in viewpoint. Images are extracted from the ZuBud database [135].

While we may assume that the building facades are presented in a clean, isolated, and full manner, the facade that we have to recognize in an image is often in the context of multiple other building facades, and/or trees, possibly occluding each other. The building facade may also be partially shifted out of the image as we can see in Figure 1.2. This is very common for images of building facades since the camera is unlikely to capture the entire facade at a close distance.

1.2 Approaches in the Thesis

There are two fundamentally different strategies to tackle the above-mentioned matching problems in urban scenes. The first aims to eradicate the false matches as much as possible through some verification steps. With the remaining of only/mainly the true correspondences, the matching results are optimized. The alternative is to try finding a plausible way to obtain much more correspondences and concurrently no/fewer false matches. The difficulty is the change of the camera position that will induce an apparent deformation of the image scene. Thus, matching must be invariant with respect to such deformations. These are described in more detail below.

1.2.1 Enhancing Image Matches through Verification

Many of the computer vision approaches and applications [22, 89, 118, 130, 156, 177] involve some hypothesize-and-verify techniques [41, 110, 113, 150, 152, 153] to optimize image matches by reducing false matches. However, many of these techniques (e.g., RANSAC [41]) are problematic when the percentage of false matches is large. It has been shown in [19, 71, 175] that true correspondences can significantly improve recognition tasks and therefore the elimination of false matches is crucial. In this thesis, we explore several geometric and appearance-based methods to eliminate false matches despite the large percentage of false matches. We take into account the resemblances between correspondences, e.g., their textures, patterns, Euclidean distances and angles among neighbors, etc.

1.2.2 Invariant Matching across Different Views

As illustrated earlier in Figure 1.2, the deformation of an image patch gets larger when the view angle gets wider. State-of-the-art image matching techniques such as SIFT [89] (in Chapter 2) that are based on region patches are ineffective when the view angle is wide.

None of the existing techniques can handle large viewpoint changes very well, e.g., for view angles that are higher than 40° [99]. To this end, we utilize some of the geometric representations that we can find in man-made scenes to explore on the problem. For example, based on observation, we learnt that the facades of many structures in man-made environments comprise some planar quadrilaterals, e.g., buildings, windows, doors, etc. Detecting these man-made features narrows down our search for interest/salient features to match in the entire image, as we do not want to match features like trees, grass, sky, sand, etc. In this thesis, the geometric matching techniques to obtain invariance to different views are considered.

1.3 Organization of the Thesis

The thesis is organized as follows. Chapter 2 discusses background concepts related to feature-based/object/scene recognition techniques, namely the three key processes that are involved - feature extraction, feature description, and feature matching.

In Chapter 3, we present a verification model that can be used to enhance image matching results by eliminating false correspondences. Several geometric and appearance-based measurements are exploited, and conditional probability is used to compute the probability of each true correspondence.

In Chapter 4, we propose an alternative to hypothesize and verify salient keypoints and true correspondences in urban scene images using an image pattern verification technique. We novelly combine the local invariant features and traditional corner-like features to select salient keypoints with invariant and corner-like properties, which are highly distinctive and well-suited for matching images of urban environments. Keypoints that do not comprise any corner-like patterns are rejected. Matches are verified by eliminating correspondences that have dissimilar pattern responses which are most likely false matches.

Chapter 5 describes an approach that is dedicated to the matching of building facades in urban scenes across widely separated views. We extract hypotheses of building facades based on a robust line fitting algorithm. Each hypothesis is defined by a planar convex quadrilateral in the image, which we call a “q-region”. A projective transformation model can be derived from each pair of q-regions in the two images. If the pair of q-regions is correctly aligned, all line segments and interest points within them will fit the transformation model and match accordingly. Consequently, the largest pair of correctly aligned q-regions will produce the largest list of matches.

In Chapter 6, a fully affine invariant region descriptor coined “PRIUS” is being derived. Interest points and their neighboring points within close spatial proximity are used to describe the detected quadrilateral regions, and to robustly align building facades through their projectivity. We explore the applicability of PRIUS to a mobile guide application.

Lastly, we conclude this thesis with a summary of the main ideas and future work in Chapter 7. Our list of publications is included at the end of the thesis.

Chapter 2

Literature Review

Chapter Overview. This chapter encompasses a detailed description of some of the state-of-the-art object and scene recognition techniques. These references cover work that are most closely related to the wide-baseline matching process that is at the core of this thesis.

2.1 Introduction

Wide-baseline matching across two images typically involves three processes - feature extraction, feature description, and feature matching - which will be discussed in this chapter. The first is the extraction of features, or so called points of interest, in the two images and the selection of a region around each point of interest. It then follows by associating a descriptor to each region. Correspondences may thus be established by applying some matching algorithms to the descriptors [98].

In computer vision, we have to understand the difference between the two types of features - global and local features. Recognitions using global features are usually solved by image segmentation techniques which are still lacking in robustness and generalization. Existing segmentation techniques often have to be combined with domain knowledge in order to effectively solve an image segmentation problem for a problem domain.

On the other hand, local features refer to the representations of image contents at interest regions on the images. They are local as they are related to small regions on objects instead of the whole object itself. This property makes them distinctive as well as robust to occlusion and clutter. Local features for an object or scene can be matched to a database of local features each representing an unique object. Differences between these images can be a substantial range of affine distortion, noise level, change in illumination, scaling, rotation, and change in viewpoint. The local features should be invariant to these differences in order to robustly match the two images of the same object or scene. Invariance of a local feature refers to its ability to tolerate these differences. A good local feature should be highly distinctive in the way that a single feature can be correctly matched with high probability against a large database of features from many images. However the more invariant a feature is, the less distinctive it will be, which is the trade-off between invariance and distinctiveness.

Local features have been very popular and widely used in applications such as stereo-vision [10], image/video retrieval [40, 53, 96, 137], image registration [158, 173], robot localization [18, 89, 114], object recognition [17, 87], scene classification [20, 39, 144], texture recognition [70], and building panoramas [3, 22]. In this thesis, we will only focus and discuss on the local features.

2.2 Feature Extraction

Feature extraction is typically used as the first step of many vision tasks such as image matching and recognition, localization, SLAM (Simultaneous Localization And Mapping) [79], and tracking. Since an image's resolution can be extremely high, it is not practical to take every pixel of the image as a feature and describe it by a vector. We can just extract certain features and infer the contents of an image from them. These features can be called points of interest, which are the points in an image that have well-defined

positions and can be robustly detected. This means that a point of interest can be a corner, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal. In the literature, “corner”, “point of interest”, “interest point”, “keypoint”, “key feature”, “salient feature”, and “feature point” are used somewhat interchangeably. One of the most important requirements for these points is that they can be differentiable from their neighboring image points. If this is not the case, it will not be possible to match them uniquely with other corresponding points in another image. There are numerous feature extraction techniques proposed in the past, and a few of them will be discussed in details as follows - the Moravec detector [103, 104], Harris/Plessey detector [54], Harris-Laplacian detector [95], Harris-Affine detector [96, 97], Difference of Gaussians (DoG) [87, 89], Maximally Stable Extremal Regions (MSER) [92], and the Features from Accelerated Segment Test (FAST) [122, 123] and FAST-Enhanced Repeatability (FAST-ER) [124] detectors.

2.2.1 Moravec detector

This is one of the earliest corner detectors developed. It is proposed by Moravec in 1977 for his research involving the navigation of the Stanford Cart [103, 104] through a clustered environment. The corner detector defines interest points as point where there is a large intensity variation in every direction, which is the case at corners. Intensity variation is measured by placing a square window (e.g., 3×3 , 5×5 , or 7×7 pixels) centered at a particular position, and then shifting this window by one pixel in each of the eight principle directions (horizontally, vertically, and two diagonals). The intensity variation for a given shift is calculated by taking the Sum of Squared Differences (SSD) in intensity of corresponding pixels between the two windows. A lower number indicates greater similarity. If the window is in a region of uniform intensity, then the nearby patches will look similar, and therefore, shifting the window in any direction will result

in only a small intensity variation. For a window straddling a corner, or an isolated pixel, will result in a large intensity variation for all shift directions. Let a grayscale 2-dimensional (2D) image be given by I , the square window over the area be positioned at (x, y) and shifting it by (u, v) , then the weighted SSD between the two windows, denoted S , is given by

$$S(u, v) = \sum_{x, y} w(x, y) (I(x + u, y + v) - I(x, y))^2. \quad (\text{Eq. 2.1})$$

The corner strength is defined as the $\min\{S\}$ between the patch and its neighbors (the eight shift directions). If this number is locally maximal, then an interest point is found.

However, Moravec detector suffers from a number of problems. It is not rotationally invariant as the response is anisotropic. Intensity variation is only calculated at a discrete set of shifts (the eight principle directions). If an edge is present that is not in the direction of the neighbors, then it will not be detected as an interest point. This causes the corner detector to have a poor repeatability rate. Moreover, it has a noisy response due to a binary window function. It responds too readily to edges since any imperfections in an

Figure 2.1: Image results [117] from the Moravec detector. In (a), the Moravec operator is applied to an image of blocks. In (b), the Moravec operator is applied to an image comprising the partial view of a house.

edge due to noise, pixelation, or intensity quantization may result in the local minimum intensity variation over all shift directions being relatively large. The Harris detector [54] is proposed to address many of these limitations, which will be discussed next. Some image results obtained by the Moravec detector are shown in Figure 2.1.

2.2.2 Harris/Plessey detector

The Harris detector [54], sometimes also known as the Harris and Stephens or the Plessey detector, is probably the most widely used point detector in the vision literature. It improves upon the Moravec detector by addressing its limitations. It takes the differential of the corner score, referred to as autocorrelation [179], with respect to direction directly instead of using shifted patches like in the Moravec detector. The result is a far more desirable detector in terms of detection and repeatability rate but at the cost of requiring significantly more computation time. The Harris detector also solves the anisotropic problem of the Moravec detector by using a smooth circular window, for example, a 2D Gaussian function,

$$w(x, y) = \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right). \quad (\text{Eq. 2.2})$$

Given the weighted SSD between two patches, denoted S ,

$$S(u, v) = \sum_{x, y} w(x, y) (I(x, y) - I(x + u, y + v))^2, \quad (\text{Eq. 2.3})$$

the shift $I(x + u, y + v)$ can be approximated by a Taylor expansion [7]. Let I_u and I_v be the partial derivatives [2] of I such that

$$I(x + u, y + v) \approx I(x, y) + I_u(x, y)u + I_v(x, y)v. \quad (\text{Eq. 2.4})$$

This produces the approximation

$$S(u, v) \approx \sum_{x, y} w(x, y) (I_u(x, y)u + I_v(x, y)v)^2, \quad (\text{Eq. 2.5})$$

which can be written in matrix form,

$$S(u, v) \approx \begin{pmatrix} u & v \end{pmatrix} \mathbf{M} \begin{pmatrix} u \\ v \end{pmatrix}. \quad (\text{Eq. 2.6})$$

$\mathbf{M}$ is the structure tensor (second-moment matrices) defined as

$$\mathbf{M} = \sum_{x,y} w(x, y) \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix} = \begin{bmatrix} \langle I_u^2 \rangle & \langle I_u I_v \rangle \\ \langle I_u I_v \rangle & \langle I_v^2 \rangle \end{bmatrix}, \quad (\text{Eq. 2.7})$$

where the angle brackets denote averaging. This matrix is called a Harris matrix. An interest point, or rather a corner, is characterized by a large variation of S in all directions of the vector $\begin{pmatrix} u & v \end{pmatrix}$. The Moravec detector responds too readily to edges as only the minimum of S is taken into account. The Harris detector however reformulates the corner measure to ensure the variation of S with the direction of shift. The strength of the corner can be determined by the second derivative of S . This is done by considering the eigenvalues [6] (λ_1 and λ_2) of $\mathbf{M}$. Based on the magnitudes of the eigenvalues, the following inferences can be made:

- (i) If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, then there is no feature of interest in this pixel (u, v) .
- (ii) If $\lambda_1 \approx 0$ and λ_2 is a large positive value, then an edge is found.
- (iii) If λ_1 and λ_2 are both large distinct positive values, then a corner is found.

However, it is useful to have a measure of the corner and edge quality or response instead of just the corner and edge classification regions. The size of the response will be used to select isolated corner pixels and to thin the edge pixels. The measure of the corner response is

$$R = \text{Det}(\mathbf{M}) - k(\text{Trace}(\mathbf{M}))^2, \quad (\text{Eq. 2.8})$$

$$\text{where } \text{Det}(\mathbf{M}) = \lambda_1 \lambda_2,$$

$$\text{Trace}(\mathbf{M}) = \lambda_1 + \lambda_2,$$

$$\text{and } k = \text{constant}.$$

Using $Det(\mathbf{M})$ and $Trace(\mathbf{M})$ in the formulation avoid the explicit and computational expensive eigenvalue decomposition of $\mathbf{M}$. The value of k has to be determined empirically. In the literature, the values in the range 0.04 to 0.15 have been reported as feasible. The final step is to threshold on the value of R and perform non-maximal suppression to find local maxima. A good (corner) point should have a large intensity change in all directions, i.e. R should be a large positive value. Some image results provided by the Harris detector are shown in Figure 2.2.

Figure 2.2: Image results [117] from the Harris detector. In (a), the Harris detector is applied to an image of blocks. In (b), the Harris detector is applied to an image comprising the partial view of a house.

The Harris detector has been shown to have good rotational and illumination invariance in addition to identifying the interest points in the image [133]. Nevertheless, the Harris detector also has its limitations. It is computationally demanding mainly because it requires convolution with a Gaussian window. It is also sensitive to noise since it relies on gradient information. The effect of noise can be reduced by increasing the size of the Gaussian window as this allows more pixels to be considered when calculating the gradient. However, increasing the size of the Gaussian window will increase the computational

demand of the algorithm.

Moreover, the Harris detector has poor localization on many junction types. In a performance assessment of feature detectors by Rockett [121], the Harris detector is the worst performing detector in localization accuracy. However, in term of repeatability as a performance measure, Schmid et al. [133] find that Harris detector consistently gives the most repeatable set of detections. To obtain scale invariance, Mikolajczyk et al. [95] introduce the Harris-Laplacian detector which combines the Harris detector with the idea of a Gaussian scale-space representation in order to create a scale-invariant detector. This will be discussed in the next section.

2.2.3 Harris-Laplacian detector

The scale-space representation is a set of images represented at different levels of resolution [171]. Mikolajczyk et al. [95] introduce the Harris-Laplacian detector which first computes a set of scale-space representation images for the Harris corner detector. It then select points where the normalized Laplacian is maximal over scales. The amplitude of the spatial derivatives, in general, decreases with scale, hence in order to maintain the property of scale invariance, the derivative function must be normalized with respect to the scale of observation. The Harris corner detector is modified such that it can be applied over scale-space. The detector analyzes the autocorrelation [179] matrix $\mathbf{M}$ of every location in an image that is computed from normalized image derivatives,

$$\mathbf{M} = \sigma_D^2 g(\sigma_I) \otimes \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}, \quad (\text{Eq. 2.9})$$

where $g(\sigma_I)$ is the Gaussian kernel of scale σ_I , and $\mathbf{x}$ is a point (x, y) . $L(\mathbf{x})$ is the Gaussian-smoothed image, and the $\otimes$ operator denotes convolution. $L_x(\mathbf{x}, \sigma_D)$ and $L_y(\mathbf{x}, \sigma_D)$ represent the image derivative in their respective direction computed over an image by convolving with a Gaussian kernel of scale σ_D . The image derivatives are

normalized by multiplying with σ_D^2 where σ_D is proportional to the scale of the target image. After building a scale-space representation for the Harris function, the candidate points are subjected to scale maxima detection. For each of the candidate points found on different levels, it is verified if it forms a maximum in the scale direction. The Laplacian F of a point $\mathbf{x}$ is defined as

$$F(\mathbf{x}, \sigma_D) = |\sigma_D^2 (L_{xx}(\mathbf{x}, \sigma_D) + L_{yy}(\mathbf{x}, \sigma_D))|, \quad (\text{Eq. 2.10})$$

where derivatives are computed with uniform Gaussian filters. Candidate point $\mathbf{x}$ at scale σ_{Dn} is maximum in Laplacian in scale direction if the following condition is satisfied,

$$F(\mathbf{x}, \sigma_{Dn}) > F(\mathbf{x}, \sigma_{Dn-1}) \wedge F(\mathbf{x}, \sigma_{Dn}) > F(\mathbf{x}, \sigma_{Dn+1}), \quad (\text{Eq. 2.11})$$

where σ_{Dn-1} is a sampled scale just smaller than σ_{Dn} , and σ_{Dn+1} is a sampled scale just larger than σ_{Dn} . Figure 2.3 shows the scale-space representation for two images with points detected using the Harris-Laplacian detector. There are many point-to-point correspondences between the different resolution levels.

Figure 2.3: Image results [95] from the Harris-Laplacian detector where points are detected on different resolution levels. The scale ratios corresponding to the real scale changes between the images are indicated by the pointers.

The Harris-Laplacian detected points are scale invariant and work well for isotropic regions that are viewed from similar viewing angle. It is not invariant to arbitrary affine transformations and viewpoints. The Harris-Affine detector is proposed to address these limitations, and will be discussed next.

2.2.4 Harris-Affine detector

The Harris-Affine detector [96, 97] can be seen as an extension of the Harris-Laplacian detector. The detector can identify similar elliptical regions in images taken from different viewpoints that are related by simple geometric transformation, e.g., scaling, rotation, and shearing. These detected regions have been called both invariant and covariant. For initialization, the approximate location and scale of each interest point are extracted by the multi-scale Harris detector. An iterative procedure is then applied to each point which modifies the location, scale, as well as shape of the point neighborhood. This converges the interest point to a stable point that is invariant to affine transformation. The following procedure [97] is applied for a given initial interest point $\mathbf{x}^{(0)}$:

- (i) Initialize $U^{(0)}$ to an identity matrix.
- (ii) Normalize window $W(\mathbf{x}_w) = I(\mathbf{x})$ centered on $U^{(k-1)}\mathbf{x}_w^{(k-1)} = \mathbf{x}^{(k-1)}$.
- (iii) Select integration scale σ_I at point $\mathbf{x}_w^{(k-1)}$.
- (iv) Select differentiation scale $\sigma_D = s\sigma_I$, which maximizes $\frac{\sigma_{min}(\mu)}{\sigma_{max}(\mu)}$, with $s \in [0.5, ..., 0.75]$ and $\mu = \mu(\mathbf{x}_w^{(k-1)}, \sigma_I, \sigma_D)$.
- (v) Detect spatial localization $\mathbf{x}_w^{(k)}$ of a maximum of the Harris measure (Eq. 2.8) nearest to $\mathbf{x}_w^{(k-1)}$ and compute the location of the interest point $\mathbf{x}^{(k)}$.
- (vi) Compute $\mu_i^{(k)} = \mu^{-\frac{1}{2}}(\mathbf{x}_w^{(k)}, \sigma_I, \sigma_D)$.

(vii) Concatenate transformation $U^{(k)} = \mu_i^{(k)} \cdot U^{(k-1)}$ and normalize $U^{(k)}$ to $\lambda_{max}(U^{(k)}) = 1$.

(viii) Go to Step (ii) if $1 - \lambda_{min}(\mu_i^{(k)})/\lambda_{max}(\mu_i^{(k)}) \geq \text{small error } \epsilon_c$.

Mikolajczyk and Schmid [97] had good success with $\epsilon_c = 0.05$. Some image results obtained by the Harris-Affine detector are shown in Figure 2.4. The points used for initialization that were detected by the multi-scale Harris detector are shown in column (a). There is a significant change in location between these points detected at different scales, and the circles in the corresponding images (top and bottom row) do not cover the same image regions. The affine invariant points which the initial points converge to are presented in column (b). The method converges correctly even if the location and scale of the initial points are relatively far from the point of convergence. Convergence generally

Figure 2.4: Image results [96] from the Harris-Affine detector. The circles around the points show the scales of detection. In (a), initial interest points are detected by the multi-scale Harris detector. In (b), points and corresponding affine regions are obtained after applying the iterative algorithm. In (c), point neighborhoods are normalized with the estimated matrices to remove stretch and skew.

takes less than 10 iterations. The minor differences between the regions in column (b) are caused by the imprecision of the scale estimation and the error ϵ_c . Column (c) shows the points normalized with the estimated matrices to remove stretch and skew. The regions that correspond between the two images (top and bottom row) are visibly shown.

In an analysis done by Mikolajczyk et al. [99], it shows that the Harris-Affine detector has a reasonable robustness to viewpoint changes. The detector maintains a repeatability score of above 50% up until a viewpoint angle of above 40-degree. It detects a high number of repeatable and matchable regions even under large viewpoint changes. The detector also remains very consistent under scale changes of above 2.8, image blurring, or illumination changes.

2.2.5 Difference of Gaussians (DoG)

In computer vision, the Difference of Gaussians (DoG) is an image enhancement algorithm that works on grayscale images. It involves the subtraction of one blurred version of an original image from another less blurred version of the original. The algorithm is believed to mimic how neural processing in the retina of the eye extracts details from images destined for transmission to the brain [37, 93, 174]. The DoG algorithm we discuss here is used in the Scale Invariant Feature Transform (SIFT) [87, 89] (Section 2.3.6) as a blob detector.

The DoG algorithm minimizes the cost of interest point extraction by taking a cascade filter approach in which the more expensive operations are applied only at locations that pass all prior tests. The first stage is to identify locations and scales that can be repeatedly detected under differing views of the same object. Locations that are invariant to the scale change of the image can be detected by searching for stable features across all possible scales using a continuous function of scale known as scale space [171]. Koenderink [67] and Lindeberg [83] have shown that the Gaussian function is the only possible scale-space kernel under a variety of reasonable assumptions. Therefore, the scale space of an

image, which is defined as a function $L(x, y, \sigma)$, is produced from the convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with an input image $I(x, y)$ as

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y), \quad (\text{Eq. 2.12})$$

where $\otimes$ is the convolution operator, and

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x^2+y^2)/2\sigma^2}. \quad (\text{Eq. 2.13})$$

The scale-space extrema in the DoG function, $D(x, y, \sigma)$, is computed from the difference between two nearby scales of filtered images $L(x, y, \sigma)$, separated by a constant multiplicative factor k given by

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) \otimes I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (\text{Eq. 2.14})$$

The scale space of the input image $D(x, y, \sigma)$ is prepared in a way illustrated in Figure 2.5.

The DoG function provides a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2 \nabla^2 G$, as proven by Lindeberg [83]. Therefore, finding the extrema in DoG space is approximately equivalent to finding the extrema in Laplacian space. To detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared with the pixels of all its 26 neighbors (i.e. 8 neighbors in the current image, 9 neighbors in the scale above, and 9 neighbors in the scale below) as shown in Figure 2.6. This point is an extrema if its value is the minimum or maximum among its 26 neighbors. This point is called a keypoint in the SIFT framework. The advantage of searching for keypoints over a range of scales is that both small and large keypoints can be detected. Small keypoints help to solve occlusion problems, while large keypoints contribute to the robustness of the system towards noise and image blur.

Figure 2.5: Scale-space in Difference of Gaussian (DoG) [89]. For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images (left). Adjacent Gaussian images are subtracted to produce the DoG images (right). After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

Figure 2.6: Detecting Maxima and minima in DoG space [89]. Maxima and minima of the DoG images are detected by comparing a pixel with its 26 neighbors (marked with circles) in 3×3 regions at the current and adjacent scales.

In Lowe’s method [89], he also attempted to eliminate unstable keypoints with low contrast or are poorly localized along an edge. Keypoints with low contrast are rejected as they are sensitive to noise, and keypoints localized along the edge are also rejected as they in general do not make significant difference with nearby points. First, for each keypoint, interpolation of nearby data is used to determine its position. The interpolation is achieved by using the quadratic Taylor expansion [7] of the DoG scale-space function $D(x, y, \sigma)$ with the keypoint as the origin. The Taylor expansion is defined as

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}, \quad (\text{Eq. 2.15})$$

where D and its derivatives are evaluated at the keypoint, and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum, denoted as $\hat{\mathbf{x}}$, is determined by taking the derivative of this function with respect to $\mathbf{x}$ and setting it to zero, giving

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}}. \quad (\text{Eq. 2.16})$$

If the offset $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, it indicates that the extremum lies closer to another keypoint. In this case, the keypoint is changed and the interpolation is performed about that point instead. Otherwise, $\hat{\mathbf{x}}$ will be added to its keypoint to get the interpolated estimation for the location of the extremum. The function value at the extremum, $D(\hat{\mathbf{x}})$, is used for rejecting unstable extrema with low contrast by substituting Eq. 2.16 into Eq. 2.15, giving

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}}. \quad (\text{Eq. 2.17})$$

In Lowe’s experiments [89], all extrema with a value of $|D(\hat{\mathbf{x}})|$ less than 0.03 are discarded.

The next step is the elimination of keypoints that have poorly determined locations but have high edge responses. For poorly defined peaks in the DoG function, there is a large principle curvature across the edge but a small curvature in the perpendicular

direction. The principal curvatures can be computed from a 2×2 Hessian matrix [84, 95], $\mathbf{H}$, at the location and scale of the keypoint, giving

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}. \quad (\text{Eq. 2.18})$$

The eigenvalues of $\mathbf{H}$ are proportional to the principal curvatures of D . The derivatives D_{xx} , D_{xy} , and D_{yy} , can be estimated by taking the differences between neighboring sample points. Thus, the ratio of the two eigenvalues reflects whether the interest point is on the edge or not. To simplify the solution, we only need to verify if the ratio of the principal curvatures is below some threshold r in the condition

$$\frac{\text{Trace}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}, \quad (\text{Eq. 2.19})$$

$$\text{where } \text{Trace}(\mathbf{H}) = D_{xx} + D_{yy},$$

$$\text{and } \text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2.$$

In Lowe's experiments [89], all keypoints that have a ratio between the principal curvatures greater than 10 are discarded. Figure 2.7 shows the entire stages of keypoint selection.

Figure 2.7: Image results [170] from the DoG detector, showing stages of keypoint selection. In (a), the scale space extrema are detected. In (b), low contrast keypoints are discarded. In (c), keypoints that are located on edges are also filtered out.

2.2.6 Maximally Stable Extremal Regions (MSER)

Maximally Stable Extremal Regions (MSER) is proposed by Matas et al. [92] in 2002 as a method of blob detection in images. It extracts from an image a number of covariant regions called MSER. Each MSER is a stable connected component of some level sets of the image. Optionally, elliptical frames can be attached to the MSER by fitting ellipses to the regions. Its definitions [92] are as follows:

Image I is a mapping $I : D \subset \mathbb{Z}^2 \rightarrow S$. Extremal regions are well defined on images if:

- (i) S is totally ordered, i.e. reflexive, antisymmetric and transitive binary relations $\leq$ exists.
- (ii) An adjacency (neighborhood) relation $A \subset D \times D$ is defined.

Region Q is a contiguous subset of D , i.e. for each $p, q \in Q$ there is a sequence $p, a_1, a_2, \dots, a_n, q$ and $pAa_1, a_iAa_{i+1}, a_nAq$, where A denotes “adjacent to”.

(Outer) Region Boundary $\partial Q = \{q \in D \setminus Q : \exists p \in Q : qAp\}$, i.e. the boundary ∂Q of Q is the set of pixels being adjacent to at least one pixel of Q but not belonging to Q .

Extremal Region $Q \subset D$ is a region such that for all $p \in Q, q \in \partial Q : I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region). This means that all the surrounding pixels have a higher or lower intensity than the region.

Maximally Stable Extremal Region (MSER). Let $Q_1, \dots, Q_{i-1}, Q_i, \dots$ be a sequence of nested extremal regions, i.e. $Q_i \subset Q_{i+1}$. Extremal region Q_{i^*} is maximally stable if and only if $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i|$ has a local minimum at i^* , where $|\cdot|$ denotes cardinality. $\Delta \in S$ is a parameter of the method.

Extremal regions have two important properties:

- (i) Continuous (and thus projective) transformation of image coordinates, and thus it is affine invariant.

- (ii) Monotonic transformation of image intensities, and thus it is invariant to photometric changes.

Since the regions are defined exclusively by the intensity function in the region and the outer border, MSER provides many key characteristics of the regions which make them very useful. Over a large range of thresholds, the local binarization is stable in regions having the 5 properties as follows:

- (i) Invariance to affine transformation of image intensities.
- (ii) Covariance to adjacency preserving (continuous).
- (iii) Stability: only regions whose support is nearly the same over a range of thresholds is selected.
- (iv) Multi-scale detection without performing any smoothing. However, Forssén and Lowe [44] assert that the detection of MSER in a scale pyramid improves repeatability and number of correspondences across scale changes.
- (v) The set of all extremal regions can be enumerated in worst-case $O(n)$, where n is the number of pixels in the image [115].

Some image results obtained by the MSER detector are shown in Figure 2.8.

In an analysis done by Mikolajczyk et al. [99], it shows that MSER has a repeatability of 92% on the region density test, and outperforms the other detectors in many of the images under the viewpoint change and light change tests. However, MSER proves to be most sensitive to image blur during the image blurring test. In addition, as pointed out by Morel and Yu [107], MSER is not fully scale invariant as its scale normalization is based on the size (area) of the detected extremal regions. Nevertheless, scale change is not just a homothety. It involves a blur, and is followed by sub-sampling. The blur

Figure 2.8: Image results from the MSER detector where ellipses are fitted to the detected regions. Experiment was done on two building images of different views from the ZuBud database [135].

merges the regions and changes their shapes and sizes. In some of the later work, MSER has been adapted to color images. Forssén and Lowe [44] replace thresholding of the intensity function with agglomerative clustering based on color gradients. Donoser and Bischof [36] use MSER to track color objects by performing MSER detection on the Mahalanobis distance [91] to a color distribution.

2.2.7 FAST and FAST-ER detectors

Many detectors define corners using a model or an algorithm and apply it directly to the image. An alternative is to exploit some machine learning methods to train a classifier on the model and then apply it to the image. For instance, a multi-layer perceptron can be trained on sample corners from some models and applied to the image after some processing [28, 33]. The FAST (Features from Accelerated Segment Test) detector [122, 123] is one detector that is derived from machine learning. It considers a circle of 16 pixels around the corner candidate p . The original detector [122, 125] classifies p as a

corner if there exists a set of n contiguous pixels in the circle which are all brighter than $I_p + t$ or all darker than $I_p - t$, where I_p is the intensity of the candidate pixel and t is a threshold. Figure 2.9 illustrates this. n is chosen to be 12 as it admits a high-speed test which can be used to exclude many non-corners. The test examines the four pixels at the four compass directions - 1, 5, 9, and 13 - as shown in Figure 2.9. In order to consider p as a corner, at least three of these pixels have to be brighter than $I_p + t$ or darker than $I_p - t$. The full segment test criterion can then be applied to the remaining candidates by examining all pixels in the circle.

Figure 2.9: Corner hypothesis in an image patch by the FAST detector [123]. The pixel at p is the center of a candidate corner. The 16 highlighted squares are pixels forming a circle-like shape around p . The arc is indicated by the dashed line that passes through 12 contiguous pixels which are brighter than p by at least a threshold t .

The work is then expanded to use machine learning to address the points [123]. Two stages are involved. First, the 16 pixel locations on the circle around each pixel are extracted from a set of images from some target application domain using the discussed test criterion for n and a suitable threshold. For each location on the circle $x \in \{1 \dots 16\}$, the pixel at that position relative to p (denoted as $p \rightarrow x$) can have one of 3 states:

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t & \text{(darker);} \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t & \text{(similar);} \\ b, & I_p + t \leq I_{p \rightarrow x} & \text{(brighter).} \end{cases} \quad (\text{Eq. 2.20})$$

Let P be the set of pixels in all the training images. Choosing an x partitions P into three subsets, P_d , P_s , and P_b , where

$$P_b = \{p \in P : S_{p \rightarrow x} = b\}, \quad (\text{Eq. 2.21})$$

and P_d and P_s are defined in a similar fashion. Stage 2 exploits the ID3 algorithm [119] to select the x which yields the most information about whether the candidate pixel is a corner, measured by the entropy of K_p where K_p is a boolean variable (i.e. true if p is a corner and false otherwise). The total entropy of K for an arbitrary set of corners, Q , is

$$H(Q) = (c + \bar{c}) \log_2(c + \bar{c}) - c \log_2 c - \bar{c} \log_2 \bar{c}, \quad (\text{Eq. 2.22})$$

$$\text{where } c = |\{i \in Q : K_i \text{ is true}\}| \quad (\text{number of corners}),$$

$$\text{and } \bar{c} = |\{i \in Q : K_i \text{ is false}\}| \quad (\text{number of non corners}).$$

The choice of x then yields the information gain,

$$H_g = H(P) - H(P_d) - H(P_s) - H(P_b). \quad (\text{Eq. 2.23})$$

After selecting the x which yields the most information, the process is applied recursively on all three subsets, i.e. x_b is selected to partition P_b into $P_{b,d}$, $P_{b,s}$, $P_{b,b}$, x_s is selected to partition P_s into $P_{s,d}$, $P_{s,s}$, $P_{s,b}$, and so on. Each x is chosen to yield maximum information about the set it is applied to. The process terminates when the entropy of a subset is zero. This means that all the p in this subset have the same value of K_p , i.e. they are either all corners or all non-corners. This will definitely occur since K is an exact function of the learning data. Consequently, a decision tree will be created which can correctly classify all corners seen in the training set and therefore correctly embodies the rules of the chosen FAST corner detector. Some image results provided by the FAST detector (with $n = 9$) are shown in Figure 2.10.

Experiments [123, 124] show that the best value of n is 9. The detector has a high level of repeatability under large aspect changes and for different kinds of feature. It is

(a)

(b)

Figure 2.10: Image results from the FAST detector (with $n = 9$). Experiment was done on two building images of different views from the ZuBud database [135].

also much faster than other existing corner detectors such as Harris [54] and DoG [89] detectors in tests [123, 124]. However, the detector is not robust to high level noise. In addition, it is dependent on a threshold.

The later version of the detector proposed in 2010 is known as a FAST-ER detector [124], where the additional “ER” stands for “Enhanced Repeatability”. The detector is generalized by defining a feature detector to be a ternary decision tree which detects points with high repeatability. The repeatability of such a detector is a non-convex function of the configuration of the tree, and therefore the tree is optimized using simulated annealing. This results in a multi-objective optimization. The cost function for the tree is defined as

$$k = \left(1 + \left(\frac{w_r}{r}\right)^2\right) \left(1 + \frac{1}{N} \sum_{i=1}^N \left(\frac{d_i}{w_n}\right)^2\right) \left(1 + \left(\frac{s}{w_s}\right)^2\right), \quad (\text{Eq. 2.24})$$

where r is the repeatability, d_i is the number of detected corners in frame i , N is the number of frames, and s is the size (number of nodes) of the decision tree. The effect of these costs are controlled by w_r , w_n , and w_s . For efficiency, repeatability is computed

at a fixed threshold as opposed to a fixed number of features per frame. To obtain invariance to rotation, reflection, and intensity inversion while reducing the burden on the optimization algorithm, each time the tree is evaluated, it is applied sixteen times: at four simple rotations (90° apart), with all combinations of reflection and intensity inversion. Experiments in [124] show that the FAST-ER detector has a dramatic improvement in repeatability over the FAST detector, especially in noisy images.

2.3 Feature Description

An extracted key feature, defined in terms of a specific structure in the image data, can be represented in different ways. For example, a set of extracted key features can be represented as a boolean variable in each image point that describes whether a corner is present at that point. Alternatively, we can define a specific region around the corner point and either represent it in terms of the average color (three scalars) or a color histogram (three functions). There are many alternatives to represent the key features. In computer vision, the choice of feature representation can be a critical issue. In some cases, a higher level of detail in the description of a feature may be necessary for solving the problem, however this comes at the cost of having to deal with more data and more demanding processing. In some applications, it is not sufficient to extract only one type of feature to obtain the relevant information from the image data. Instead, two or more different features are extracted, resulting in two or more feature descriptors at each image point. A common practice in computer vision is to organize the information provided by all these descriptors as the elements of one single vector, commonly referred to as a feature vector.

We will now discuss some of the existing feature representation algorithms - SIFT [87, 89], PCA-SIFT [66], Shape Context [14, 17], GLOH [99], SURF [11, 13], Affine-SIFT [107,

176], and simple grayscale patches. In this discussion, an instance of a feature representation is referred to as a feature descriptor.

2.3.1 Scale Invariant Feature Transform (SIFT)

The SIFT descriptor [87, 89] is represented relative to one or more orientations that are assigned to each keypoint based on local image gradient directions. This is the key step in achieving invariance to rotation. In Lowe’s method [87, 89], he uses the Difference of Gaussians (DoG) as discussed in Section 2.2.5 to detect keypoints such that the descriptor is invariant to scale changes. The Gaussian-smoothed image $L(x, y, \sigma)$ at the keypoint’s scale σ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L(x, y)$ at scale σ , the gradient magnitude $m(x, y)$, and orientation $\theta(x, y)$ are pre-computed using pixel differences:

$$\begin{aligned} m(x, y) &= \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \\ \theta(x, y) &= \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \end{aligned} \quad (\text{Eq. 2.25})$$

The magnitude and direction calculations for the gradient are done for every pixel in a neighboring region around the keypoint in the Gaussian-blurred image L . An orientation histogram with 36 bins covering the full 360 degrees range of orientation is formed. Each sample in the neighboring window that is added to a histogram bin is weighted by its gradient magnitude, and also by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. The peaks in this histogram correspond to dominant orientations of local gradients. Once the histogram is filled, the orientations corresponding to the highest peak and all local peaks that are within 80% of the highest peaks are assigned to the keypoint. In the case of multiple orientations being assigned, additional keypoints are created having the same location and scale as the original keypoint for each additional orientation.

Next, a descriptor vector is computed for each keypoint such that the descriptor is highly distinctive and partially invariant to illumination and viewpoint changes. This step is performed on the image closest in scale to the keypoint's scale. Figure 2.11 illustrates the computation of the SIFT descriptor. First, the image gradient magnitudes and orientations are sampled around the keypoint location using the scale of the keypoint to select the level of Gaussian blur for the image. The coordinates of the descriptor and the gradient orientations are then rotated to align with the keypoint orientation to obtain orientation invariance. For efficiency, the gradients are pre-computed for all levels of the pyramid which are illustrated with small arrows at each sample location on the left side of Figure 2.11.

Figure 2.11: Computation of SIFT descriptor [89]. The descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location as shown on the left. These are weighted by a Gaussian window indicated by the circle. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions as shown on the right with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2×2 descriptor array computed from an 8×8 set of samples, whereas the experiments by Lowe [89] use 4×4 descriptors computed from a 16×16 sample array.

The magnitude of each sample point is then weighted by a Gaussian function with σ equal to one half the width of the descriptor window. This is illustrated with a circular

window on the left of Figure 2.11. The Gaussian window is employed to avoid sudden changes in the descriptor with small changes in the position of the window, and also to give less emphasis to gradients that are far from the center of the descriptor as these are most affected by registration errors.

The keypoint descriptor allows for significant shift in gradient positions by creating orientation histograms over 4×4 sample regions. As shown on the right of Figure 2.11, there are 8 directions for each orientation histogram with the length of each arrow corresponding to the magnitude of that histogram entry. A gradient sample on the left can shift up to 4 sample positions while still contributing to the same histogram on the right, thereby achieving the objective of allowing for larger local positional shifts.

The descriptor is formed from a vector comprising the values of all the orientation histogram entries corresponding to the lengths of the arrows on the right of Figure 2.11. The best results are reported [89] to be achieved with a 4×4 array of histograms with 8 orientation bins, and a total of $4 \times 4 \times 8 = 128$ element feature vector for each keypoint. An image result of location recognition in a complex scene using SIFT descriptor is shown in Figure 2.12.

Finally, the descriptor is normalized to unit length to enhance invariance to affine changes in illumination. To reduce the effects of non-linear illumination, a threshold of 0.2 is applied and the vector is re-normalized.

Mikolajczyk et al. [98] evaluated a variety of local descriptors and identified SIFT and SIFT-based descriptors as being the most resistant to common image deformations. These descriptors exhibit the highest matching accuracies (recall rates) for an affine transformation of 50 degrees. After this transformation limit, results start to become unreliable. For scale changes in the range 2 to 2.5 and image rotations in the range 30 to 45 degrees, SIFT and SIFT-based descriptors again outperform other local descriptors with both textured and structured scene content.

Figure 2.12: Image result [89] of location recognition in a complex scene using SIFT descriptor. The query images for four locations are shown at the upper left, and the 640×315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the bottom image with keypoints shown as squares, and an outer parallelogram showing the boundaries of the query images under the affine transform used for recognition.

2.3.2 Principle Components Analysis-SIFT (PCA-SIFT)

Principle components analysis (PCA) [60] is a standard technique used to reduce multi-dimensional data sets to lower dimensions for analysis. It has been applied to a broad class of computer vision problems, including feature selection (e.g., [48]), object recognition (e.g., [108]), and face recognition (e.g., [154]). Ke and Sukthankar [66] demonstrate that PCA is well-suited to representing keypoint patches once these patches have been transformed into canonical scales, positions, and orientations. The PCA-SIFT descrip-

tor [66] is an alternate representation for local image descriptors of the SIFT algorithm described in the previous section. It accepts the same inputs (i.e. sub-pixel location, scale and orientations) as the SIFT descriptor. It can be summarized in the following steps:

- (i) Locate a 41×41 patch at the given scale centered over the keypoint.
- (ii) Rotate the patch to align its dominant orientation to a canonical direction. For keypoints with multiple dominant orientations, a representation for each orientation is built in the same manner as SIFT.
- (iii) Pre-compute an eigenspace to express the gradient images of local patches. It can be pre-computed by applying PCA to the gradient patches in a set of training images.
- (iv) Compute the local image gradient of a given patch.
- (v) Create a vector by concatenating both horizontal and vertical gradient maps.
- (vi) Normalize the vector to unit magnitude to make it illumination invariant.
- (vii) Project the gradient image vector using the eigenspace created in (iii) to derive a compact feature vector.
- (viii) The Euclidean distance between two feature vectors can be used to determine whether two vectors correspond to the same keypoint in different images.

As implied by the steps above, the PCA-SIFT descriptor is invariant to orientation changes and partially invariant to illumination changes just like the way the SIFT descriptor is. It has also been shown to have similar accuracy to SIFT in recognition [99], but more compact (i.e. lower dimension) and therefore leading to improvements in running time.

2.3.3 Shape context

Just like the SIFT descriptor discussed in Section 2.3.1, shape contexts [14, 16, 17] also create histograms, but they are based on edges to describe shapes that allow for measuring shape similarity and the recovering of point correspondences [14]. Each shape context is a 3D histogram of edge point locations and orientations. Edge points refer to a set of points sampled from the shape contours of the targeted object using an edge detector. Shape context uses bins that are uniform in log-polar space to emphasize close-by local structures, making the descriptor more sensitive to positions of nearby sample points than points further away. Figure 2.13 shows the computation of a histogram using log-polar coordinates. Figure 2.14 shows the bin counts in the histogram indicated by the grayshade (black = large, white = small).

For a point p_i on the shape, the histogram h_i is computed by simply counting the number of edge points within a bin given by

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \text{bin}(k)\}. \quad (\text{Eq. 2.26})$$

This histogram is defined to be the shape context of p_i . Shape context is invariant to changes in illumination since it is a histogram computed from edge points. Scale invariance can be obtained by normalizing all radial distances by the mean distance α between all point pairs in the shape [16, 17] although the median distance can also be used [14, 15]. To make it invariant to orientation, the feature detector has to assist in aligning the dominant orientation of the local patch to a canonical direction. Shape contexts have been demonstrated to be robust to deformations, noise, and outliers [15] using synthetic point set matching experiments [29]. Unfortunately, shape features are rarely adequate for discriminatory object recognition of 3D objects from arbitrary viewpoints [35]. In an analysis done by Mikolajczyk et al. [98], it showed that in textured scenes or in scenes where edges are not reliable, shape context obtains a low score.

Figure 2.13: Log-polar histogram bins used in computing shape contexts [17]. Belongie et al. [17] use 5 bins for quantizing distance $\log r$ between the rest of the edge points from the given edge point, and 12 bins for quantizing the angle θ between them. Compact representation of point distribution relative to each bin is shown.

Figure 2.14: Shape contexts of three edge points [17]. Darker bins indicate that larger number of edge points are located inside the bins. The first and second shape contexts are quite similar as they are for two closely related points, while the third shape context is very different.

2.3.4 Gradient Location and Orientation Histogram (GLOH)

The GLOH descriptor [98] is an extension of the SIFT descriptor discussed in Section 2.3.1. Similar to the SIFT descriptor, the GLOH descriptor describes the gradient orientations of the image patches. Instead of sampling the gradient orientations in a rectangular grid, the GLOH descriptor samples them in a log-polar location grid similar to the one used in shape context described in Section 2.3.3. The histogram of each interest point consists of 3 bins in radial direction (radius set to 6, 11, and 15) and 8 in angular direction, which results in 17 location bins with 16 orientation bins in each. This gives a 272 bin histogram. Principle Components Analysis (PCA) [60] is then applied to reduce the dimension of the GLOH descriptor to 128 (i.e. only the 128 largest eigenvectors are used).

Mikolajczyk et al. [98] present the experimental evaluation of interest region descriptors in the presence of real geometric and photometric transformations. In most of the tests, the GLOH descriptor obtained the best results, followed closely by the SIFT descriptor. This shows the robustness and the distinctive character of the region-based descriptors.

2.3.5 Speeded Up Robust Features (SURF)

The SURF descriptor, proposed by Bay et al. [11, 13], is a more recent descriptor after SIFT, PCA-SIFT, and GLOH discussed previously. It makes use of similar properties of the SIFT descriptor which are the relative strengths and orientations of gradient information. It describes a distribution of Haar-wavelet responses within the interest point neighborhood.

The first step is to fix a reproducible orientation for the interest points. In the original scheme [11, 13], interest points are detected using a Fast-Hessian detector. The assigned orientation makes the descriptor invariant to rotation. For this purpose, Haar-wavelet

Figure 2.15: Haar wavelet types and square regions used for SURF descriptor. Left: Haar wavelet types used for SURF [13]. Right: Square regions of different scales centered around interest points of a Graffiti scene [13].

responses are calculated in both the x and y directions as shown in Figure 2.15, within a circular neighborhood of radius $6s$ around the interest point, where s is the scale at which the interest point was detected. The sampling step size is scale dependent and is chosen to be s . Wavelet responses are also computed at that current scale s . Integral images for fast filtering are used as the size of the wavelets is big at large scales.

After calculating the wavelet responses, they are weighted with a Gaussian $\sigma = 2.5s$ centered at the interest point. The responses are then represented as vectors in a space with the horizontal response strength along the abscissa and the vertical response strength along the ordinate. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window covering an angle of $\frac{\pi}{3}$. The horizontal and vertical responses within the window are summed to yield a new vector, and the longest such vector lends its orientation to the interest point.

The second step is to construct a square region centered around the interest point, aligned to the selected orientation in the previous step. The size of the window is $20s$. Figure 2.15 illustrates the square regions. The regions are split up regularly into smaller

4×4 square sub-regions to keep important spatial information in. In addition, a few simple features at 5×5 regularly spaced sample points are computed for each of these sub-regions. The Haar wavelet responses d_x (in horizontal direction) and d_y (in vertical direction) are summed up over each subregion to form a first set of entries to the feature vector. “Horizontal” and “vertical” are defined in relation to the selected interest point orientation (see Figure 2.16). The responses d_x and d_y are first weighted with a Gaussian ($\sigma = 3.3s$) centered at the interest point to increase the robustness towards geometric deformations and localization errors.

Figure 2.16: The formulation of the SURF descriptor [11]. An oriented quadratic grid with 4×4 square sub-regions is laid over the interest point (left). For each square, the wavelet responses are computed. The 2×2 sub-divisions of each square correspond to the actual fields of the descriptor. These are the sums d_x , $|d_x|$, d_y , and $|d_y|$, computed relatively to the orientation of the grid (right).

The sum of the absolute values of the responses $|d_x|$ and $|d_y|$ are also extracted in order to observe information about the polarity of the intensity changes. Hence, each sub-region will have a four-dimensional descriptor vector $\mathbf{v}$ for its underlying intensity structure

$$\mathbf{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right). \quad (\text{Eq. 2.27})$$

A descriptor vector is formed for all 4×4 sub-regions of length 64. The wavelet responses are invariant to any bias in illumination (offset). Finally, the descriptor is converted into a unit vector in order to achieve invariant to contrast (a scale factor). Figure 2.17 shows the properties of the SURF descriptor for three distinctively different image intensity patterns within a sub-region. Combinations of such local intensity patterns result in a distinctive descriptor.

Figure 2.17: Properties of the SURF descriptor for three distinctively different image intensity patterns within a sub-region [11]. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

The SURF descriptor is similar in concept to the SIFT descriptor as they both focus on the spatial distribution of gradient information. However, the SURF descriptor integrates the gradient information within a sub-patch, whereas the SIFT descriptor depends on the orientations of the individual gradients. This makes the SURF descriptor less sensitive to noise. Bay et al. [13] also claimed that SURF outperforms other previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed much faster. For object detection, its performance has been illustrated in [12]. Cattin et al. [27] use SURF for mosaicing human retina images, a task that no other detector/descriptor scheme was able to cope with.

2.3.6 Affine-SIFT (ASIFT)

ASIFT [107, 176], proposed in 2009, simulates all possible distortions caused by a variation of the camera optical axis direction. It then follows by applying the SIFT method described in Section 2.3.1. An overview of the ASIFT algorithm is shown in Figure 2.18.

Figure 2.18: Overview of the ASIFT algorithm [107]. The square images A and B represent the compared images. ASIFT simulates all distortions caused by a variation of the camera optical axis direction. The simulated images, represented by the parallelograms, are then compared by SIFT which is invariant to scale changes, rotation, and translation.

To put it more explicitly, ASIFT simulates three parameters: the scale, the camera longitude angle (i.e. equivalent to the pan angle) and the latitude angle (i.e. equivalent to the tilt angle), and normalizes the other three (i.e. translation and rotation). In the work [107, 176], the key observation is that although a tilt distortion is irreversible due to its non-commutation with the blur, it can be compensated up to a scale change by digitally simulating a tilt of same amount in the orthogonal direction. As

opposed to other normalization methods (e.g., by SIFT [89]) that suffer from this non-commutation, ASIFT is able to achieve full affine invariance. ASIFT proceeds by the following steps [176]:

- (i) Each image is transformed by simulating all possible affine distortions caused by the change of orientation of the camera axis. These distortions are dependent on the longitude ϕ and latitude θ parameters. The images undergo ϕ -rotations followed by tilts with parameter $t = \left\lfloor \frac{1}{\cos \theta} \right\rfloor$. For digital images, the tilt is performed as a t sub-sampling and requires an anti-aliasing filter in the direction of x , namely the convolution by a Gaussian with standard deviation $c\sqrt{t^2 - 1}$ where $c = 0.8$ [106].
- (ii) The rotations and tilts are performed for a finite and small number of latitudes and longitudes. The sampling steps of these parameters ensure that the simulated images stay close to any possible view generated by other values of ϕ and θ .
- (iii) All simulated images are compared to each other by SIFT [89]. All camera parameters are either normalized or simulated by ASIFT.
- (iv) The simulated latitudes θ are sampled so that the associated tilts follow a geometric series $1, a, a^2, \dots, a^n$, with $a > 1$. The choice $a = \sqrt{2}$ is a good compromise between accuracy and sparsity, and the value n can go above 5. As a consequence transitional tilts going up to 32 degrees or more can be explored.
- (v) The longitudes ϕ for each tilt t follow an arithmetic series $0, b/t, \dots, kb/t$, where $b \simeq 72^\circ$ is a good compromise and k is the greatest integer such that $kb/t < 180^\circ$.
- (vi) Experiments [107] show that ϕ can vary from 0 to 2π , but simulating $\phi \in [0, \pi)$ is enough to cover all possible affine transforms. It has been proven that the sampling range $t_{max} = 4\sqrt{2}$ (i.e. corresponding to a view angle $\theta_{max} = \arccos 1/t_{max} \approx 80^\circ$)

allows ASIFT to be invariant to transitional tilt as large as $(4\sqrt{2})^2 = 32$ degrees [107]. With higher resolution images, larger transition tilts would be attainable. The sampling steps $\Delta t = \sqrt{2}$ and $\Delta\phi = \frac{72^\circ}{t}$ have also been validated by applying SIFT between images with simulated tilts and longitude variations equal to the sampling step values [107].

- (vii) Complexity: Each tilt is a t sub-sampling dividing the image area by t . The number of rotated images for each tilt is $(180/72)t = 2.5t$. Thus, the algorithmic complexity is proportional to the number of tilts. Controlling the total area of the simulated images is equivalent to controlling the algorithm complexity. Indeed, the SIFT search time and memory size are proportional to the image area. This complexity can be further reduced by (a) sub-sampling the query and search images; (b) identifying the successful pairs (t, ϕ) ; (c) going back to the original resolution only for these pairs. For example, if the query and search images are sub-sampled by a 3×3 factor for the low-resolution ASIFT, the overall comparison complexity can be up to twice the SIFT complexity [176]. However, there are proposed SIFT accelerations [43, 66, 78] that can be directly applied to ASIFT.

ASIFT image matching performance has been compared with the state-of-the-art approaches in [107], using the detectors DoG [89] (Section 2.2.5), MSER [92] (Section 2.2.6), Harris-Affine and Hessian-Affine [96, 97] (Section 2.2.4). All of them are combined with the most popular SIFT descriptor [89] (Section 2.3.1). On average, ASIFT finds 20 times or more matches than any of the above approaches in the viewpoint change test.

2.3.7 Simple grayscale patches

This is one of the simplest feature descriptor, yet it is very useful and commonly used in computer vision, e.g., for template matching [24]. It samples pixels within a template

(e.g., a small square window) around each chosen or detected feature point in a grayscale image. The template acts as a descriptor, of size 5×5 , 9×9 , 15×15 , or whichever size that is appropriate in the application. To find the best match within a search image, we simply move the center of the template over each point in the search image, and calculate the similarity between the coefficients in the template and search image over the whole area spanned by the template. The similarity measurements (acquired using template matching techniques) between the template and search image will be further discussed in Section 2.4.2.

The advantages of using the simple grayscale descriptor are, first - the simplicity of construction of the descriptor; and second - the spatial information of the region is maintained. However, as this simple grayscale descriptor samples square pixels as what they are in the image, it is not rotation invariant and is highly sensitive to viewpoint and illumination changes. Improvements to the descriptor exist in the literature, either by concentrating in illumination changes [82] or in geometrical variations [52]. Examples of fast rotation invariant template matching with an exhaustive search are described in [45, 157].

2.4 Feature Matching

Once key features are extracted and described from two or more images, the next step is to establish some preliminary feature matches between these images. The feature matching technique we exploit depends partially on the application, e.g., different techniques may be preferable for matching images that are known to overlap (e.g., in panoramic image stitching [23]) vs. images that may have no correspondence at all (e.g., object recognition from a database of images [127]). In some cases, an additional verification step may be necessary, depending on the matching techniques and applications.

We will now discuss some of the feature matching techniques used in matching the local descriptors mentioned in the previous sections. They are divided into three categories - distance measures [126], template matching [24], and homography matching [118].

2.4.1 Distance measures

Distance measures have been widely adopted in comparing the local descriptors. They are classified into two categories [126] - the bin-by-bin dissimilarity measures and the cross-bin measures. Bin-by-bin dissimilarity measures only compare the contents of corresponding histogram bins. For example, given two histograms $H = (h_1, h_2, \dots, h_n)$ and $K = (k_1, k_2, \dots, k_n)$, the bin-by-bin dissimilarity measures compare h_i and k_i for all i , but not h_i and k_j for $i \neq j$. On the contrary, the cross-bin measures also contain terms that compare the non-corresponding bins.

The main drawback of the bin-by-bin measures is their sensitivity to bin size. A bin size that is too large will not have sufficient discriminative power, e.g., features that are not so close will fall into the same bin and considered to be the same. On the other hand, a bin size that is too small will place close features into two bins (on two sides of the bin boundary) which will never be matched. Therefore, a slight shift of the bin size would cause an abrupt change to the descriptor [90]. Cross-bin measures do not have this boundary problem, nevertheless they have their own demerits. For example, a common problem of the quadratic distance measure [112] (Section 2.4.1.4) is the overestimation of the mutual similarity, resulting in false positives in retrieval results.

Some of these distance measures will be discussed as follows, (1) bin-by-bin dissimilarity measures - Minkowski distance [126], histogram intersection [147], and chi-square statistic [169]; (2) Cross-bin measures - Quadratic distance [112] and match distance [136, 165].

2.4.1.1 Minkowski distance

The Minkowski distance [126] is a metric on Euclidean space which can be considered as a generalization of both the Manhattan distance and the Euclidean distance [168]. The Manhattan distance is often used for computing dissimilarity between color images [147], while the Euclidean distance is usually used to find the distance between two points. It was shown in [143] that for image retrieval, the Manhattan distance results in many false negatives because neighboring bins are not considered. Hence, Euclidean distance is one of the most common bin-by-bin distance measures used in comparing local descriptors such as SIFT [89], GLOH [98], and PCA-SIFT [66] that was discussed earlier in Section 2.3. The Minkowski distance of order p (p -norm distance) is defined as

$$d_p(H, K) = \left(\sum_{i=1}^n |h_i - k_i|^p \right)^{1/p}, \quad (\text{Eq. 2.28})$$

where $p > 1$, otherwise the triangle inequality will not hold. If p is 1, it is known as the Manhattan distance. If p is 2, it is the Euclidean distance (2-norm distance). In the limit where p approaches infinity, it is known as the Chebyshev distance.

2.4.1.2 Histogram intersection

Histogram intersection [147] is a bin-by-bin distance measure which adds up the minimum values between each pair of corresponding histogram bins in two images. If there is a large overlap between the bin values of two corresponding histograms, then the minimum values will also be large. Therefore, two similar histograms will result in a large intersection value; or in other words, the lesser the area that is left, the closer the match is between the two histograms. The histogram intersection distance is defined as

$$d(H, K) = 1 - \frac{\sum_{i=1}^n \min(h_i, k_i)}{\sum_{i=1}^n k_i}. \quad (\text{Eq. 2.29})$$

Histogram intersection is well-known for its ability to handle partial matches when the areas of the two histograms are different. It is adopted by the color histogram indexing

approach proposed by Swain and Ballard [146]. In [147], Swain and Ballard show that when the areas of the two histograms are equal, the histogram intersection is equivalent to the normalized Manhattan distance discussed in Section 2.4.1.1.

2.4.1.3 Chi-square statistic

The chi-square (χ^2) statistic [169] is a nonparametric statistical technique used to determine if a distribution of observed frequencies (O) differs from the theoretical expected frequencies (E) [4]. It uses nominal (categorical) or ordinal level data, thus the test uses frequencies rather than using means and variances. Generally, the χ^2 statistic summarizes the discrepancies between the expected and observed numbers of times each outcome occurs by summing the squares of the discrepancies, normalized by the expected numbers over all categories [4]. The formula for computing χ^2 is

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}. \quad (\text{Eq. 2.30})$$

The observed frequencies O_i in this case can be our h_i and the expected frequencies E_i can be $\frac{h_i+k_i}{2}$. For a perfect histogram match, h_i can be expected to be equal to k_i . Therefore the formula can be written as

$$\chi^2 = \sum_{i=1}^n \frac{(h_i - m_i)^2}{m_i}, \quad (\text{Eq. 2.31})$$

where $m_i = \frac{h_i+k_i}{2}$. This bin-by-bin distance measure is adopted by the shape context descriptor [17] discussed in Section 2.3.3.

2.4.1.4 Quadratic distance

The quadratic distance is used for color indexing in the QBIC (Query By Image Content) project [112]. It is a cross-bin distance measure which does not enforce a one-to-one correspondence between mass elements in the two histograms. Ling and Okada [86] have recently shown that the quadratic distance measure can improve on the SIFT feature

matching approach [89]. This distance measures the weighted similarity between histograms, which provides more desirable results than “like-bin” only comparisons. The quadratic distance is defined as

$$d(H, K) = \sqrt{(h - k)^T \mathbf{A}(h - k)}, \quad (\text{Eq. 2.32})$$

where h and k are vectors that list all the entries in H and K . Cross-bin information is incorporated via a similarity matrix $\mathbf{A} = [a_{ij}]$, where a_{ij} denotes the similarity between bins i and j , defined as

$$a_{ij} = 1 - \frac{dist_{ij}}{dist_{\max}}. \quad (\text{Eq. 2.33})$$

i and j are sequential (scalar) indices into the bins. $dist_{ij}$ is the ground distance between bins i and j of the histogram, and $dist_{\max} = \max(dist_{ij})$.

Although in general the quadratic distance is not a true distance, it can be shown that with a proper choice of $\mathbf{A}$, the quadratic distance is indeed a distance [126]. Since the quadratic distance computes the cross similarity between features, it can be computationally expensive. In addition, it was shown in [143] that using the quadratic distance in image retrieval results in higher false positives as it tends to overestimate the mutual similarity of color distributions without a pronounced mode.

2.4.1.5 Match distance

The match distance [136, 165] between two one-dimensional (1D) histograms is defined as the Manhattan distance (Section 2.4.1.1) between their corresponding cumulative histograms. The distance is computed as

$$d(H, K) = \sum_{i=1}^n |\hat{h}_i - \hat{k}_i|, \quad (\text{Eq. 2.34})$$

where $\hat{h}_i = \sum_{j \leq i} h_j$ is the cumulative histogram of H , and similarly for K . The match distance does not extend to higher dimensions as the relation $j \leq i$ is not a total ordering in more than one dimension (1D), and the resulting arbitrariness causes problems [126].

2.4.2 Template matching

Several algorithms have been developed [59, 145] for feature tracking as well as for the initial estimation of the position of a given template pattern in an image. A common approach to achieve this is template matching [24]. This approach has been applied to various situations such as to detect edges in images [162], to navigate a mobile robot [69], as a part of quality control in manufacturing [5], and to detect calcified nodules within digital chest X-rays [1].

Template matching is a well-known method for finding small portions of a source image which match a template image. The matching process usually involves moving the template to all possible positions in a larger source image, and computing a numerical index that indicates how well the template matches the image in that position. The match is done on a pixel-by-pixel basis.

Correlation is a statistical technique used in template matching to show how strongly a pair of images (i.e. template and source image) is correlated. It is the most direct way to find similarity between the two grayscale image patches (Section 2.3.7) by computing a similarity score. However, correlation is known for its high dimensionality which results in a high computational complexity for recognition. Works have been proposed to speed up the process, e.g., by Ke and Sukthankar [66], using the image gradient patch and applying PCA [60] to reduce the size of the descriptor. We will now discuss two of the commonly used correlation techniques - Cross-correlation [167] and Normalized Cross-Correlation (NCC) [82].

2.4.2.1 Cross-correlation

Cross-correlation [167] estimates the degree to which the source image and template are correlated. Let $f(x, y)$ denotes the intensity value of the source image f of size $M_x \times M_y$ at the point (x, y) , $x \in \{0, \dots, M_x - 1\}$, $y \in \{0, \dots, M_y - 1\}$, and t as a template of size

$N_x \times N_y$. A common way to calculate the position of the pattern in the image f is to evaluate the cross-correlation value at each point (u, v) for f and the template t , which has been shifted by u steps in the x direction and by v steps in the y direction. The basic definition for the cross-correlation coefficient is given by

$$c(u, v) = \sum_x \sum_y f(x, y) t(x - u, y - v). \quad (\text{Eq. 2.35})$$

The cross-correlation is similar in nature to the convolution of two functions.

There are several limitations for using cross-correlation for template matching. The range of $c(u, v)$ is dependent on the size of the template. In addition, it is not invariant to changes in image intensity amplitude such as those caused by changing lighting conditions across the image sequence.

2.4.2.2 Normalized Cross-Correlation (NCC)

Normalized Cross-Correlation (NCC) [82] overcomes part of the limitations in cross-correlation by normalizing the image and template vectors to unit length, yielding a cosine-like correlation coefficient. To calculate the correlation between the source image and template, NCC is a reasonable choice in many cases [82].

The NCC equation is calculated by

$$\gamma(u, v) = \frac{\sum_x \sum_y [f(x, y) - \bar{f}_{u,v}] [t(x - u, y - v) - \bar{t}]}{\left\{ \sum_x \sum_y [f(x, y) - \bar{f}_{u,v}]^2 \sum_x \sum_y [t(x - u, y - v) - \bar{t}]^2 \right\}^{\frac{1}{2}}}, \quad (\text{Eq. 2.36})$$

where $\bar{f}_{u,v}$ denotes the mean value of $f(x, y)$ within the region under the template t , and similar notation $\bar{t}$ denotes the mean value of the template t . The denominator in Eq. 2.36 is the variance of the zero mean image function $f(x, y) - \bar{f}_{u,v}$, multiplied by the variance of the shifted zero mean template function $t(x - u, y - v) - \bar{t}$. Due to this normalization, $\gamma(u, v)$ is independent to changes in brightness or contrast of the image, which are related to the mean value and the standard deviation. Nevertheless, NCC is computationally expensive.

In order to minimize the computational cost, a fast algorithm known as the “Fast Normalized Cross-Correlation” [82] for obtaining normalized cross-correlation from transform domain convolution has been introduced.

2.4.3 Homography matching

Homography [118] can very much simplify matching for sets of feature points by reducing the search complexity, while providing an invariant affinity measure for each feature point. Any two images of the same planar surface in space are related by a homography. Suppose the homography exactly represents the mapping between image points in the two images, then all the feature points mapped from one image to the other would coincide exactly with their correspondences. This homography mapping defines a planar mapping between two camera views that have a degree of overlap [31, 55, 141, 142] as illustrated in Figure 2.19.

Figure 2.19: Homography mapping of a point on one plane to another plane.

The image coordinates for the first and second camera views are (x, y) and (x', y')

respectively, given that

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \quad (\text{Eq. 2.37})$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}. \quad (\text{Eq. 2.38})$$

At least four corresponding points are needed to derive the homography model. Singular Value Decomposition (SVD) is commonly used for computing the homography [31, 55]. The homography matrix can be written in vector form, given by

$$\mathbf{H}_m = [h_{11} \ h_{12} \ h_{13} \ h_{21} \ h_{22} \ h_{23} \ h_{31} \ h_{32} \ h_{33}]^T. \quad (\text{Eq. 2.39})$$

Hence, each pair of corresponding points $((x, y), (x', y'))$ between two camera views results in two equations (Eq. 2.37 and Eq. 2.38) in terms of the coefficients of the homography matrix $\mathbf{H}_m$ in Eq. 2.39. Eq. 2.37 and Eq. 2.38 can be rearranged to

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \end{bmatrix} \mathbf{H}_m = 0, \quad (\text{Eq. 2.40})$$

$$\begin{bmatrix} 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix} \mathbf{H}_m = 0. \quad (\text{Eq. 2.41})$$

For n corresponding points, a $2n \times 9$ matrix $\mathbf{Q}$ can be constructed and then used to minimize $\|\mathbf{QH}_m\|$, subject to the constraint $\|\mathbf{H}_m\| = 1$. The value of $\mathbf{H}_m$ can then be estimated by using SVD.

However in reality, the homography is only an approximation of the mapping as the surface in space may not be a perfect planar. In this case, mapped feature points do not coincide exactly. If the homography is computed from only four corresponding points, this disparity generally increases with distance from the basis used. Another difficulty of doing homography matching is the finding of the initial few corresponding points (at least four) in the two images to derive the homography model. The few corresponding points have to be correct as homography is extremely sensitive to outliers [77].

2.5 Chapter Summary

In this chapter, we have explored through background studies of several existing feature extraction, feature description, and feature matching techniques. It is clear that there are many different techniques to tackle the object recognition problem. In our case, we are bringing the problem a step further by addressing this problem in urban environments.

We shall see many of the discussed state-of-the-art techniques, (e.g., DoG (Section 2.2.5), MSER (Section 2.2.6), SIFT (Section 2.3.1), ASIFT (Section 2.3.6), NCC (Section 2.4.2.2), etc.) in the coming chapters. They are employed either as part of our approach, or for result comparisons in the experiments.

In the coming Chapters, we are proposing verification approaches to enhance the correspondence matching results by eliminating false matches, and also investigating new ways to increase the number of point correspondences regardless of the viewing variations.

Chapter 3

Enhancing Image Matches via a Verification Model

Chapter Overview. In the literature, we have seen a boom in wide-baseline matching approaches proposed for locating correspondences between images. However, wrong correspondences or so called outliers are still rather inevitable, especially in urban environments where there exist presence of repetitive structures, and/or a large dissimilarity in viewpoints. In this chapter, we propose a verification model to enhance the image matching results by significantly reducing the number of outliers. Several geometric and appearance-based measurements are exploited, and conditional probability is used to compute the probability of each true correspondence. The model is validated by extensive experiments on images from the ZuBud database, which are taken in different weather conditions, seasons and with different cameras.

3.1 Introduction

One of the most challenging tasks in computer vision is the locating of true or reliable correspondences between images. It has been shown that true correspondences can significantly improve recognition tasks [19, 71]. However, locating of true correspondences without the presence of outliers is difficult, especially in urban environments where the

repetitive structures and dissimilarity in viewpoints frequently exist [75, 132]. Existing approaches cannot effectively deal with this outlier problem. The existence of outliers is not trivial given that a single pair of false correspondence is enough to generate erroneous results, e.g., in operations like homography which is extremely sensitive to noise and outliers [77].

Throughout the years, there has been an inspiring body of work on improving the matching performance by extending local features to be invariant to full affine transformations (e.g., by Baumberg [10]; Brown and Lowe [21]; Matas et al. [92]; Mikolajczyk and Schmid [96]; Obdržálek and Matas [116]; Pritchett and Zisserman [118]; Tuytelaars and Van-Gool [156]). However, none of them are fully affine invariant or robust to large viewpoint changes, as many of them start with initial feature scales and locations selected in a non-affine invariant manner due to the prohibitive cost of exploring the full affine space. Perspective projection distortion is the inevitable misrepresentation of the 3-dimensional (3D) space when projected onto a 2-dimensional (2D) surface. It is impossible to accurately depict 3D reality on a 2D plane. Nevertheless, we can approximate it to a certain degree by assuming that the perspective projection distortion is small. Correspondences cannot be established by only comparing appearance-based regions of fixed shapes like rectangles or circles, since these shapes are not preserved under affine transformation.

Mikolajczyk and Schmid [96] averred that the repeatability of affine features is lower than scale-invariant features unless the affine distortion is greater than a 40-degree tilt of a planar surface; also wider affine invariance may not be important for many applications as training/reference views are best taken at least after every 30-degree rotation in viewpoint in order to capture non-planar changes and occlusion effects for 3D objects. In this chapter, we take this discernment and try to corroborate correspondences that have reasonable geometry differences (e.g., angle and distance) within their neighbors in two

corresponding images. Besides these geometry factors, we also look at some important appearance-based factors, e.g., the texture and pattern differences in each correspondence. We combine all these measurements by using conditional probability and a data driven re-weighting approach, which computes the probability of each true correspondence. Most, if not all, of the outliers will be effectively eliminated at this stage. As this work targets on urban scene images (e.g., buildings) whose facades are mainly planar, we follow up with RANSAC homography [38, 55] to confirm the true correspondences. RANSAC homography works very well right after outlier elimination, since most/all of the outliers have been eliminated with a high percentage of inliers. We name the entire model a “verification model” [73]. We test and try various factors that affects the correspondences and select these factors as the most significant ones in obtaining good results.

3.2 Related Work

Mikolajczyk and Schmid [98] evaluated a variety of object recognition algorithms and identified that the SIFT and SIFT-based algorithms are the most resistant to common image deformations and have achieved the best performance. SIFT provides features that are invariant to image scale, translation, rotation, and partially invariant to illumination and viewpoint changes. Details on application of these features can be found in [23, 87, 88, 134]. In its original matching scheme, a pair of keypoints is considered a match if the distance ratio between the closest match and the second closest match is below a certain threshold. While the distance ratio can eliminate some of the false correspondences, we often still need to identify correct subsets of correspondences containing less than 1% inliers.

To solve the outlier problem, the RANSAC algorithm [41, 55] and other similar hypothesize-and-verify methods [110, 151, 152, 178] have been proposed in the litera-

ture. The RANSAC algorithm is a robustness method based on random sampling and rejects all correspondences not conforming to the found homography model or epipolar geometry. Although these methods work fine in many applications [22, 89, 156, 177], they perform poorly when the number of false correspondences outnumbers the number of true correspondences; or when the number of correspondences is modest (limited). The RANSAC idea was modified by Nistér [113] to include competitive verification of models. The algorithm named “Preemptive RANSAC” was demonstrated to perform well in a real-time structure-from-motion system. The limitation is that only a fixed number of models are evaluated, which is equivalent to a priori assumption that a lower bound on the fraction of inliers is known. This limits the applicability of preemptive RANSAC in wide baseline stereo where the fraction of inliers varies widely.

Another rather popular method is the Hough Transform [9, 51, 57], which clusters features in pose space. The Hough Transform identifies clusters of features with a consistent interpretation by using each feature to vote for all object poses that are consistent with the feature. However, in [89], it was shown that follow-ups are required after Hough Transform is performed in order to eliminate more outliers, e.g., least-squares pose determination, followed by a probabilistic model given in [88]. Moreover, Hough Transform requires a huge computation load in pixel transformation and a large storage (or memory) is also required for the voted Hough space. Without proper parallelization, it will be very difficult for Hough Transform to achieve real-time performance.

3.3 Verification Model

We introduce a verification model [73] which comprises several measurements to adjudicate the reliability of each correspondence pair. In our work, we apply it to the SIFT algorithm [89]. To find any resemblances between correspondences, we sample pixels within a 15×15 square window around each correspondence pair and measure their

patterns and textures. In addition, we detect reasonable Euclidean distances and angles among their neighbors. Each of these measurements will give a quantitative score to each correspondence pair. We combine all these measurements using conditional probabilities and a data driven re-weighting approach, which computes the probability of each true correspondence pair. RANSAC homography [38, 55] is used right after probability estimation to confirm true correspondences. The overall flow diagram of our approach is shown in Figure 3.1.

Figure 3.1: Overall flow diagram of our approach (SIFT + Verification model).

3.3.1 Pattern approximation

This measurement was motivated by the work of Sluzek [138] which uses moment-derived complex pattern approximations of circular patches around keypoints. Corners and corner-like patterns (e.g., junctions) are predominantly significant as they generally preserve their geometry over a wide range of radii of circular patches. Thus, in our work we implement two corner-like patterns, i.e. proper corner and T-junction. The measurements are straightforward. A pair of correspondence attains a high score if they have similar responses to the two pattern approximations. In other words, we scrutinize the difference in the quantitative responses of the correspondence windows with these pattern approximations.

Figure 3.2: Model configurations of (a) a corner, and (b) a T-junction.

The model configuration of a corner over a circle of radius R is defined by two angles β_1 and β_2 , and two intensities (or colors) as illustrated in Figure 3.2(a). Similarly, the model configuration of a T-junction consists of two angles and three intensities (or colors) is illustrated in Figure 3.2(b). Given any circular image window of radius R , the parameters of its optimum corner approximation can be found from moment-based expression m specified in [138]. The orientation angle β_2 in Figure 3.2(a) is extracted using

$$\beta_2 = \arctan 2(\pm m_{01}, \pm m_{10}), \quad (\text{Eq. 3.1})$$

while the angular width β_1 is computed from

$$\beta_1 = 2 \arcsin \sqrt{1 - \frac{16[(m_{20} - m_{02})^2 + 4m_{11}^2]}{9R^2(m_{10}^2 + m_{01}^2)}}, \quad (\text{Eq. 3.2})$$

or

$$\beta_1 = 2 \arccos \frac{4}{3} \sqrt{\frac{(m_{20} - m_{02})^2 + 4m_{11}^2}{R^2(m_{10}^2 + m_{01}^2)}}. \quad (\text{Eq. 3.3})$$

The intensities A_1 (corner) and A_2 (background) are found using

$$A_1 = \frac{m_{00}}{\pi R^2} + \frac{3(2\pi - \beta_1)(m_{10} \cos \beta_2 + m_{01} \sin \beta_2)}{4\pi R^3 \sin 0.5\beta_1}, \quad (\text{Eq. 3.4})$$

and

$$A_2 = \frac{m_{00}}{\pi R^2} - \frac{3\beta_1(m_{10} \cos \beta_2 + m_{01} \sin \beta_2)}{4\pi R^3 \sin 0.5\beta_1}. \quad (\text{Eq. 3.5})$$

The angles β_1 , β_2 , and the intensities A_1 , A_2 , of each pair of correspondence are then compared to evaluate the similarity between them.

For T-junction as shown in Figure 3.2(b), it requires three intensity parameters ($A_1 - A_3$). Its angular width β_1 and orientation angle β_2 can be computed using

$$\frac{\pi}{2} - \beta_2 - \frac{\beta_1}{2} = \frac{\arctan 2(\pm m_{02} \mp m_{20}, \pm 2m_{11})}{2}, \quad (\text{Eq. 3.6})$$

and

$$m_{01} \cos \beta_2 - m_{10} \sin \beta_2 = \pm \frac{4}{3R} \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2} \quad (\text{Eq. 3.7})$$

simultaneously. The intensities are found as solutions of the following system of linear equations

$$\begin{cases} \frac{2m_{00}}{R^2} = A_1\pi + A_2\beta_1 + A_3(\pi - \beta_1) \\ \frac{3m_{10}}{R^3} = -2A_1c_2 + A_2(c_2 - c_{2-1}) + A_3(c_2 + c_{2-1} - 2s_2) \\ \frac{3m_{01}}{R^3} = -2A_1s_2 + A_2(s_2 - s_{2-1}) + A_3(s_2 + s_{2-1} + 2c_2) \end{cases} \quad (\text{Eq. 3.8})$$

where c_x and s_x indicate the cos and sin functions of the corresponding arguments. Simple calculations can show that the results produced by Eq. 3.1 - Eq. 3.3 and Eq. 3.6 - Eq. 3.7 are invariant to linear illumination changes, and the angular width β_1 is invariant under

any similarity transformation. Extensive experiments have also attested that the results are stable under both high and low frequency noise, image texturization and partial over and under saturation of image intensities.

This method is applicable to color images as well. In fact, it may be even more flexible since the moments of color images are 3-dimensional (3D) vectors (i.e. R , G , and B components) instead of scalars. Thus, the approximation equations for color patterns will be modified correspondingly. If scalar moments are directly replaced by moment vectors, the gray-level solution remains basically unchanged. For example, Eq. 3.2 is converted into

$$\beta_1 = 2 \arcsin \sqrt{1 - \frac{16 (\|\vec{m}_{20} - \vec{m}_{02}\|^2 + 4\|m_{11}\|^2)}{9R^2 (\|m_{10}\|^2 + \|m_{01}\|^2)}}. \quad (\text{Eq. 3.9})$$

If the direct replacement of scalars by vectors is not possible in a straightforward manner (e.g., Eq. 3.7), the scalar moments will be replaced by the dominant components of vector-moments (i.e. the components with the largest magnitudes). Subsequently, Eq. 3.7 will be replaced by

$$\begin{aligned} & m_{01}(Z) \cos \beta_2 - m_{10}(Z) \sin \beta_2 \\ &= \pm \frac{4}{3R} \sqrt{\|\vec{m}_{20} - \vec{m}_{02}\|^2 + 4\|m_{11}\|^2}, \end{aligned} \quad (\text{Eq. 3.10})$$

where Z can be R , G , or B , depending on which color the value of $|m_{10}(Z)| + |m_{01}(Z)|$ is the largest. Colors of the approximations are calculated identically. We just apply Eq. 3.4, Eq. 3.5, and Eq. 3.8 separately to R , G , and B colors using the moments of the corresponding color.

3.3.2 Gabor texture

Texture is something that repeats with variation. In our work, we are trying to separate what repeats and what stays the same, and validate if each correspondence pair is made of the same “stuff”. Gabor filters are a group of wavelets. For an image patch $I(x, y)$

with size $P \times Q$, its discrete Gabor wavelet transformation is given by a convolution operation:

$$G_{mn}(x, y) = \sum_s \sum_t I(x - s, y - t) \psi_{mn}^*(s, t), \quad (\text{Eq. 3.11})$$

where s and t are the filter mask size variables, and ψ_{mn}^* is the complex conjugate of ψ_{mn} which is a class of self-similar functions generated from the dilation and rotation of the mother Gabor wavelet. m and n specify the scale and orientation of the wavelet respectively, with $m = 0, 1, \dots, M - 1$; $n = 0, 1, \dots, N - 1$, and M is the number of scales and N is the number of orientations.

After applying Gabor filters on the image patch with different orientations at different scales, an array of magnitudes of the filtered image patches is obtained as follows:

$$E(m, n) = \sum_x \sum_y |G_{mn}(x, y)|, \quad (\text{Eq. 3.12})$$

$$m = 0, 1, \dots, M - 1; n = 0, 1, \dots, N - 1.$$

These magnitudes represent the energy content at different scales and orientations of the original image patches. The following mean μ_{mn} and standard deviation σ_{mn} of the magnitude of the transformed coefficients are used to represent the homogenous texture feature of the region:

$$\mu_{mn} = \frac{E(m, n)}{P \times Q}, \quad (\text{Eq. 3.13})$$

$$\sigma_{mn} = \frac{\sqrt{\sum_x \sum_y (|G_{mn}(x, y)| - \mu_{mn})^2}}{P \times Q}. \quad (\text{Eq. 3.14})$$

A feature vector $\mathbf{f}$ (texture representation) is created using μ_{mn} and σ_{mn} as the feature components [34]. M scales and N orientations are used and the Gabor feature vector is given by

$$\mathbf{f} = (\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \dots, \mu_{(M-1)(N-1)}, \sigma_{(M-1)(N-1)}). \quad (\text{Eq. 3.15})$$

An example of the Gabor filtered results is shown in Figure 3.3, with random scales and orientations. In our work, the number of scale M and orientation N is set to 4 and

6 respectively. The filter window size of $s * t$ is chosen to be 15×15 , which is the same size as the image patches. As the size is rather diminutive, four scales are sufficient for obtaining scale invariance.

Figure 3.3: Gabor filtered images with random scales and orientations.

We use the Manhattan distance measurement discussed in Section 2.4.1.1 to measure the distance between two texture features, given by:

$$d(H, K) = \sum_{i=0}^{N-1} |h_i - k_i|, \quad (\text{Eq. 3.16})$$

where $H = \{h_0, h_1, \dots, h_{N-1}\}$ and $K = \{k_0, k_1, \dots, k_{N-1}\}$ are feature vectors of the image patches. Although reordering the feature vector in circular rotation can solve the rotation invariance problem, we recover the orientation from SIFT instead for competency in performance.

3.3.3 Euclidean distance approximation

Mikolajczyk and Schmid [96] averred that the repeatability of affine features is lower than the scale-invariant features unless the affine distortion is greater than a 40 degrees tilt of a planar surface; and wider affine invariance may not be important for many applications as training/reference views are best taken at least after every 30 degrees rotation in viewpoint in order to capture non-planar changes and occlusion effects for 3D objects. In our work, we take this discernment and try to corroborate correspondences that have reasonable Euclidean distance differences (Section 2.4.1.1) within their neighbors in the two images. The Euclidean distance $d(P, Q)$ of two points, $P = (p_x, p_y)$ and $Q = (q_x, q_y)$ in an image is computed as

$$d(P, Q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}. \quad (\text{Eq. 3.17})$$

A voting system is exploited where a vote is cast to a correspondence (P, P') in the two images if the difference in Euclidean distance between itself and a neighbor (another correspondence) (Q, Q') in the two images is below an angle of 40 degrees tilt, given by

$$(P, P') = \begin{cases} 1 & \text{if } shorter_length \leq 40_rotation \\ 0 & \text{otherwise} \end{cases} \quad (\text{Eq. 3.18})$$

$$shorter_length = \min(d(P, Q), d(P', Q')),$$

$$40_rotation = \max(d(P, Q), d(P', Q')) \times \cos 40^\circ.$$

Figure 3.4 illustrates this where a vote is cast to the correspondence. The process repeats with every other neighbor, one at a time. The consolidated vote for each correspondence is proportional to its distance voting score. The approach is observably rotation invariant and the scale is recovered from SIFT.

Figure 3.4: A simple illustration of angle change within an Euclidean distance constraint. (a) shows the front view of a face with Euclidean distance of 46 pixels between corners of left eye and lips. (b) shows the side view of the face after tilting by about 40° with Euclidean distance of 52 pixels between corners of the left eye and lips. From (a) and (b), the difference in Euclidean distance is 6 pixels between face features which is below the Euclidean distance constraint of 12 pixels (derived from Eq. 3.17 and Eq. 3.18) after a 40° rotation.

3.3.4 Angle approximation

Angle approximation is similar to Euclidean distance approximation, but rather than measuring the distance between a correspondence and each of its neighbor one at a time, it measures the angle between a correspondence and every two other neighbors at a time. Three points form two vectors a and b , and two vectors form an angle θ . In the Euclidean space, there is a strong relationship between the dot product, lengths, and angles given by

$$a \cdot b = |a||b| \cos \theta, \quad (\text{Eq. 3.19})$$

where $|a|$ and $|b|$ denote the length (magnitude) of a and b , and θ is the angle between them. Thus, the angle between two vectors can be found by

$$\theta = \arccos \left(\frac{a \cdot b}{|a||b|} \right). \quad (\text{Eq. 3.20})$$

We consider two cases of an object rotating in 3D space. The angle of two intersecting lines that are formed from the projection of an object on the image plane will increment

or decrement fastest depending on where the lines lie on when the object rotates about the Y-axis. We can look at the two cases as illustrated in Figure 3.5. As shown, the angle increases the fastest when the vectors lie on both sides of the X-axis during rotation about the Y-axis. On the contrary, the angle decreases the fastest when the vectors lie on both sides of the Y-axis during rotation about the Y-axis.

Figure 3.5: Two cases of different angle increment and decrement rates when rotating about the Y-axis. (a) shows the fastest increment of angle between two vectors when rotating about the Y-axis. (b) shows the fastest decrement of angle between two vectors when rotating about the Y-axis. Solid lines: Original vectors. Dotted lines: Rotated vectors.

From the two cases, we can approximate the smallest and largest possible angles between three points (two vectors) in an image after rotating 40 degrees by applying a transformation matrix representing rotation on the Y-axis

$$\begin{bmatrix} x' \\ y' \\ 0 \end{bmatrix} = \begin{bmatrix} \cos 40^\circ & 0 & \sin 40^\circ \\ 0 & 1 & 0 \\ -\sin 40^\circ & 0 & \cos 40^\circ \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}, \quad (\text{Eq. 3.21})$$

where x and y are the original point coordinate, and x' and y' are the 40 degrees rotated point coordinate. Eventually x' will be equal to $x \cos 40^\circ$ and y' equal to y . Therefore, the rotated angle for case 1 (see Figure 3.5(a)) and case 2 (see Figure 3.5(b)) can be derived as

$$\theta'_{max} = 2 \arctan \left(\frac{y}{x \cos 40^\circ} \right), \quad (\text{Eq. 3.22})$$

and

$$\theta'_{min} = 2 \arctan \left(\frac{x \cos 40^\circ}{y} \right), \quad (\text{Eq. 3.23})$$

respectively. Hence, if the two corresponding angles (formed by three points each) in the two corresponding images fulfilled the two case conditions (i.e. within the maximum and minimum angle constraints in Eq. 3.22 and Eq. 3.23), a vote is cast and this indicates a reliable pair of correspondence. This is illustrated in Figure 3.6. The process repeats until every possible angle made by three points is checked. This approach is notably rotation and scale invariant.

Figure 3.6: A simple illustration of angle change within the angle constraints. (a) shows the front view of a face with a 75° angle between the corners of the eyes and lips. (b) shows the side view of the face after turning by about 40° . The angle between the corners of the eyes and lips is 71° . From (a) and (b), we calculate that the maximum constraint for the 71° angle (after 40° rotation) is 86° (derived from Eq. 3.22); and the minimum constraint for the 75° angle (also after 40° rotation) is 61° (derived from Eq. 3.23). Both angles fulfill their mutual constraints.

3.3.5 Final computation of scores

To compute the final score, each of the scores is first normalized to between 0 and 1. We define Conditional Probabilities for the scores obtained from Pattern approximation S_{i1} , Gabor texture S_{i2} , Distance approximation S_{i3} , and Angle approximation S_{i4} to calculate

the probability of each true pair $Pair_i$, where i is the index of the pair. The equation is given by

$$\begin{aligned}
 P(Pair_i) &= \sum_j (P_j(Pair_i, S_{i1}, S_{i2}, S_{i3}, S_{i4})) \quad (\text{Eq. 3.24}) \\
 &= \sum_j P_j(Pair_i | S_{i1}, S_{i2}, S_{i3}, S_{i4}) P_j(S_{i1}, S_{i2}, S_{i3}, S_{i4}) \\
 &= \sum_j P_j(Pair_i | S_{i1}) P_j(Pair_i | S_{i2}) P_j(Pair_i | S_{i3}) P_j(Pair_i | S_{i4}) P_j(S_{i1}, S_{i2}, S_{i3}, S_{i4}),
 \end{aligned}$$

where $P_j(Pair_i | S_{i1}, S_{i2}, S_{i3}, S_{i4})$ is the Conditional probability of Pair i , and each $S_{i1}, S_{i2}, S_{i3}, S_{i4}$ consists of two states T or F . Therefore we have 16 states. And $P_j(S_{i1}, S_{i2}, S_{i3}, S_{i4})$ is the Prior probability. For example when $j = 1$ (corresponding to F, F, F, F), then $P_1(S_{i1}, S_{i2}, S_{i3}, S_{i4}) = P(1 - S_{i1})P(1 - S_{i2})P(1 - S_{i3})P(1 - S_{i4})$. And when $j = 16$ (corresponding to T, T, T, T), then $P_{16}(S_{i1}, S_{i2}, S_{i3}, S_{i4}) = P(S_{i1})P(S_{i2})P(S_{i3})P(S_{i4})$. As we have no knowledge about the configuration for each pair, we sum up all the possible configurations based on the probability as shown in Eq. 3.24. Pairs with low probability (currently set to < 0.65) are considered unreliable and will be eliminated. Most, if not all, of the outliers will be eliminated at this stage.

The 16 parameters in the Conditional probability table are estimated using a data driven re-weighting approach. This approach will re-adjust the weighting of the parameters in the Conditional probability table. To estimate the parameters, we adopt a semi-learning algorithm. We give labels to several matching pairs. Based on the given labels, we can estimate the parameters using linear regression [101] which minimizes the least square errors of the predicted labels and user assigned labels.

Lastly, with a higher percentage of inliers remaining, the RANSAC algorithm [41, 55] is exploited for estimating homography. A few samples (minimum 4) are randomly selected to instantiate a homography model. We use a distance threshold of 0.1 between a data point and the model to decide whether the point is an inlier or not. After several

iterations, the model that has the most number of inliers is returned. The rest of the points that do not fit the model are considered as outliers.

3.4 Experimental Results

The experiments reported here are carried out based on the ZuBud database described in [135]. The database comprises of 201 buildings. 5 images per building are acquired with variations in viewpoints, in different seasons, weather and illumination conditions, and by two different cameras. There are also some occlusions by trees and by other objects in some of the images. To evaluate the accuracy in obtaining inliers in various approaches with these difficult images, we extract 4 out of 5 images per building to be used as query images and leave 1 image as the reference image. Therefore there are a total of 804 pairs of images to perform wide-baseline matching, using the following four approaches:

- (i) SIFT standalone: Original SIFT [87] with no verification.
- (ii) SIFT + correlation: SIFT with fast normalized cross-correlation [82] for verification.
- (iii) SIFT + RANSAC: SIFT with RANSAC homography [38, 55] for verification.
- (iv) SIFT + our approach: SIFT with our verification model [73] for verification.

We use the distance ratio originally introduced by Lowe [89] which has been shown to give the best results in most cases. In order to be fair to all the approaches, matches are to be visually inspected to be considered as inliers (true matches) or outliers (false matches) since some approaches are using homography. The main objective of this chapter is to enhance the image matches by minimizing on the number of outliers. Hence in our

experiment here, we evaluate on the four approaches by comparing and examining the percentage of inliers after each match.

A set of image matching results is shown in Figure 3.7, where the number of inliers (15 inliers) is less than the number of outliers (18 outliers) after performing SIFT. It gives a percentage of 45% inliers as shown in Figure 3.7(a). SIFT and correlation for verification returns 11 inliers and 12 outliers which gives a percentage of 48% inliers as shown in Figure 3.7(b). SIFT and RANSAC for verification returns 11 inliers and 2 outliers which gives a percentage of 85% inliers as shown in Figure 3.7(c). Our approach returns 14 inliers and 0 outliers which gives a percentage of 100% inliers as shown in Figure 3.7(d). Our approach outperforms the other approaches in this pair of images of difficult scene.

Another set of image matching results is shown in Figure 3.8. SIFT returns 13 inliers and 26 outliers which gives a percentage of 33% inliers as shown in Figure 3.8(a). SIFT and correlation for verification returns 13 inliers and 19 outliers which gives a percentage of 41% as shown in Figure 3.8(b). SIFT and RANSAC for verification returns 8 inliers and 1 outlier which gives a percentage of 89% inliers as shown in Figure 3.8(c). Our approach returns 8 inliers and 0 outlier which gives a percentage of 100% inliers as shown in Figure 3.8(d). Again, our approach outperforms the other approaches in this pair of images.

Some other experimental results are presented in Table 3.1. The first column “Matched pair” shows the index of pair matches, where we have a total of 804 pairs. The subsequent columns show the four different approaches as mentioned earlier (i.e. SIFT standalone, SIFT+correlation, SIFT+RANSAC, and SIFT+our approach). We can see that our approach hits above 90% in most of the image pair matches. It amounts to an average of 98%, which outperforms the rest of the approaches. SIFT+RANSAC follows closely with an average of 91%. The algorithm slightly weakens when the percentage of outliers is large after performing SIFT. SIFT+correlation which scores an average of 71% is performing slightly better than SIFT standalone (66%).

(a) SIFT standalone: 45% inliers

(b) SIFT + correlation: 48% inliers

(c) SIFT + RANSAC: 85% inliers

(d) SIFT + our approach: 100% inliers

Figure 3.7: Matching results of difficult scenes of four different approaches.

(a) SIFT standalone: 33% inliers

(b) SIFT + correlation: 41% inliers

(c) SIFT + RANSAC: 89% inliers

(d) SIFT + our approach: 100% inliers

Figure 3.8: Other matching results of urban scenes of four different approaches.

Table 3.1: Experimental values of matched image pairs in percentage (%) of inliers; and the overall average for each of the approaches among the 804 image pair matches.

Matched pair	SIFT standalone	SIFT+ correlation	SIFT+ RANSAC	SIFT+ our approach
1	74%	85%	100%	100%
2	87%	94%	100%	100%
3	47%	61%	95%	100%
4	54%	59%	97%	97%
5	34%	43%	93%	100%
6	77%	84%	100%	100%
7	95%	99%	100%	98%
8	37%	40%	85%	100%
9	65%	74%	88%	98%
10	66%	73%	91%	100%
....				
804				
Average	66%	71%	91%	98%

3.5 Chapter Summary

In this chapter, we have presented a verification model to enhance image matches by eliminating false correspondences (or so called “outliers”). We combined geometric and appearance-based measurements, and utilized conditional probability to compute the probability of each true correspondence. We vigilantly tested various measurements and selected those measurements that are the most significant in obtaining superior results.

Experimental validation was carried out on a public database (ZuBud database) comprising various images of difficult urban scenes. A comparison with some commonly used methods (e.g., RANSAC) shows that our verification model produced better results. Our model is less sensitive to large percentage of outliers compared to other methods.

Our approach proposed in this chapter focused on eliminating/reducing outliers. Nevertheless, it does not increase the number of true correspondences after performing SIFT.

In cases where the viewing angle variation is large, the resulting number of correspondences are limited (i.e. can be less than 10). Approaches on increasing number of true correspondences are introduced in Chapters 5 and 6.

Chapter 4

Image Matching by Combining Invariant and Corner-Like Features

Chapter Overview. In this chapter, we provide an alternative to hypothesize and verify salient keypoints and true correspondences in urban scene images using an image pattern verification technique. The significance and usefulness of local invariant features and traditional corner-like features have been widely proven in the literature. We novelly combine the two types of features to select salient keypoints with invariant and corner-like properties, which are highly distinctive and well-suited for matching images of urban environments. The vast number of SIFT keypoints is reduced by rejecting those that do not comprise any corner-like patterns. We further verify the matches by eliminating correspondences that have dissimilar pattern responses which are most likely false matches.

4.1 Introduction

The use of salient features, also known as keypoints, to find correspondences across multiple images is a key step in many image processing and computer vision applications. Some of the most notable examples are panorama stitching [3, 22, 23], wide baseline matching [10, 50, 63], image retrieval [65, 163], object recognition [17, 47, 89], and object

class recognition [80, 94, 109]. These salient features are landmarks in an image which are often intuitively palpable to humans. They include corners of buildings, edges of objects, features (e.g., eyes) on human faces, etc. The traditional salient features such as edges and corners have been significantly useful and applied to many problems including tracking. We use moment-derived complex image patterns (i.e. predefined shapes and contour junctions), as discussed earlier in Section 3.3.1, to detect corner-like features which often exist in man-made environments, e.g., in building facades and windows.

In recent years, there has also been an increased interest within the content-based image retrieval community in finding new types of local features, e.g., SIFT features [89] (Section 2.3.1), which provide properties robust to changes in scale and/or affine transformations. Such invariant features have proved useful in the context of image registration [158] and object recognition [87]. In this work [72], we novelly combine these invariant features with the traditional ones (corner-like features). This combination can be used to select salient keypoints which comprise invariant and corner-like properties. These keypoints are highly distinctive points so that they can be easily distinguished from other similarly extracted points in the same or another image, and therefore improving match performance. This will be discussed in Section 4.3.1.

In addition, after matching the invariant keypoints, we further optimize the matching results by eliminating correspondence pairs that have dissimilar pattern responses which are most likely false matches. This will be discussed in Section 4.3.2.

4.2 Related Work

Many different keypoint detectors have been proposed with a wide range of definitions on what points in an image are considered interesting. Some detectors find points of high local symmetry, some find areas of highly varying texture, while others locate corner points. Corner points are interesting as they are formed from two or more edges, and

edges usually define the boundary between two different objects or parts of the same object. Work on corner detectors (e.g., Moravec [104] and Harris [54] detectors) are discussed in Section 2.2. While these detectors are called corner detectors, they are not selecting just corners, they select also any image location that has large gradients in all directions at a predetermined scale.

Moment-derived complex image patterns, which include corners and other predefined shapes, are informative as they are infrequent and provide rich description of the images. In medical imaging, there is a tendency of replacing the traditional two-step object recognition (i.e. segmentation followed by shape identification) by methods directly extracting predefined objects from grey-level images (e.g., active contours [64, 139]). Over the recent years there has been considerable attention directed towards the detection of more complex contour features in raw-data images (e.g., [32, 128]).

Recently, there has been an impressive body of work on invariant local features which have shown to be rather robust with respect to changes in scale and/or affine transformations (e.g., [10, 89, 96]). Lowe’s Scale Invariant Feature Transform (SIFT) [89], as discussed in Section 2.3.1, is one of the most popular ones. However, the vast number of detected keypoints used for matching can be rather time-consuming, and not all keypoints are equally important. In addition, false matches often exist due to the repetitive structures and/or viewpoint differences in the urban environments [73].

4.3 Combining Invariant and Corner-Like Features

We discuss the the keypoint extraction, matching, and verification process next.

4.3.1 Keypoint extraction and reduction

SIFT uses the Difference of Gaussian (DoG) extrema detector [89], as discussed earlier in Section 2.2.5, to detect keypoints that are invariant to scale changes. However, due

to the large number of extrema, there can be up to thousands of detected keypoints per image even after the process of discarding low-contrast keypoints [89]. Each of these keypoints, after sampling to a 15×15 square window, may comprise some corner-like and non corner-like patterns as shown in Figure 4.1(a) and Figure 4.1(b) respectively.

Figure 4.1: SIFT samples comprising corner-like and non corner-like patterns. Corner-like patterns are grouped in (a), and non corner-like patterns are grouped in (b).

In our approach, each of these samples is computed against four predefined complex image patterns (i.e. proper corner and T-junction), which are previously discussed in Section 3.3.1. These are common corner-like patterns that are often found in man-made objects. This is to ascertain that the sampled patch comprises at least one of these predefined corner-like patterns. If it does not, it is considered of little significance and will be rejected. Figure 4.2 illustrates this, where the two samples are tested for any corner-like patterns. We can see from the quantitative response that the sample in Figure 4.2(a) has some responses from the complex image patterns while the sample in

Figure 4.2(b) does not. The sample in Figure 4.2(a) is more distinctive in this case, which is better for matching.

Figure 4.2: Complex image pattern responses of two samples. Responses for sample (a): Corner: 0.2218; T-junction: 0.5413. Responses for sample (b): Corner: 0; T-junction: 0. Sample (b) does not comprise any corner-like patterns.

4.3.2 Keypoint matching and verification

The matching is done through an Euclidean-distance (Section 2.4.1.1) based nearest neighbor approach. To increase robustness, matches are rejected for those keypoints for which the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than some threshold τ_r ; or putting it the other way round, a pair of keypoints is considered a match if the distance ratio between the nearest neighbor distance and a second nearest neighbor distance is below τ_r , whereby

$$\frac{d^2(f, f_{1st})}{d^2(f, f_{2nd})} < \tau_r^2. \quad (\text{Eq. 4.1})$$

$f \in \mathcal{R}^n$ is the descriptor to be matched and f_{1st} and f_{2nd} are the nearest and the second nearest descriptors respectively, with d^2 denoting the square of Euclidean distance between the two descriptors. The threshold $\tau_r = 0.8$ suggested in [89] was found effective for general object recognition.

In addition, while SIFT produces invariant correspondence pairs, we verify each pair of good distance match (with complex image patterns) by scrutinizing the difference in

their responses. This can be done by sampling pixels within a 15×15 square window around the correspondence pair. If the difference in their complex pattern responses is large (e.g., above a threshold of 0.5), it is very likely that this correspondence pair is a false match, and thus it will be eliminated. Figure 4.3 illustrates an example of two sample groups with similar and dissimilar patterns. However, in some cases the samples may be difficult to visualize by human eyes.

Figure 4.3: Two sample groups of small and large complex pattern response differences. Sample group in (a) has small response differences (similar patterns), while sample group in (b) has large response differences (dissimilar patterns).

4.4 Experimental Results

We perform experiments using two data sets, with images of urban scenes comprising mainly of building facades. As explained in Section 4.3.1, distinctive keypoints are found

by selecting only those that comprise the invariant and corner-like properties. This is illustrated in Figure 4.4. We can see that those keypoints that do not comprise any corner-like patterns are usually those where the image patch are rather homogeneous. Some examples are circled in red in Figure 4.4(a). These keypoints are eliminated in Figure 4.4(b). These distinctive keypoints can improve the matching performance. In

Figure 4.4: Reduction of keypoints that do not comprise any corner-like patterns. In (a), 1326 SIFT keypoints are detected. Some of the keypoints that do not comprise any corner-like patterns are circled in red. In (b), 1008 keypoints remain after eliminating those that do not comprise any corner-like patterns.

our experiment, we take 51 images of general campus scenes for matching. 1 image is taken as the reference image while the rest of the 50 images are matched against it. The scenes are of the same building with some adjustments on the camera position at each shot, giving various viewpoint and scale differences. Figure 4.5 displays a graph of the matching results from the 50 image pairs. Most of the matches are improved using the predefined keypoints (i.e. comprising the invariant and corner-like properties).

Subsequently, the remaining keypoints are matched as described in Section 4.3.2. The matches are verified by scrutinizing the difference in the patterns or corner-like responses of each correspondence pair. If it is a true match, the complex pattern responses of the

Figure 4.5: Graph shows matching results from 50 image pairs using SIFT keypoints and predefined keypoints. Generally, predefined keypoints improved matching results.

correspondence pair will be very similar. Figure 4.6 shows the before and after image results of eliminating correspondence pairs with dissimilar complex pattern responses. We observe in Figure 4.6(b) that all false correspondence pairs are eliminated.

Figure 4.7 displays a graph of the matching results from the 50 image pairs, including the result of eliminating matches with different complex pattern responses. As shown, the matching results have been further optimized after eliminating those correspondence pairs that have different complex pattern responses.

For the second data set, we select 50 image pairs from the ZuBud database [135] which were acquired by two different cameras. Each image pair comprises urban scene photographed from different viewpoints. There are also some occlusions by trees and other objects in some images. All images are of medium resolution (640×480 pixels). Figure 4.8 displays a graph of the matching results from the 50 image pairs, including the result of eliminating matches with different complex pattern responses. Most of the matches are improved using the predefined keypoints (i.e. comprising the invariant and corner-like properties). However, the matching results are not as good as the first

(a) 102 correspondence pairs, 11 are false matches.

(b) 83 correspondence pairs remain after eliminating those with dissimilar complex pattern responses. 0 false matches.

Figure 4.6: Before and after image results of eliminating correspondence pairs with dissimilar complex pattern responses. (a) and (b) show before and after the elimination respectively.

data set after eliminating those correspondence pairs that have different complex pattern responses. After further investigation, we found that several of these image pairs have rather large viewpoint differences (e.g., for view angles that separate images higher than 40°). Our approach does not work very well in this situation since the patterns of the image patches are deformed with respect to the viewpoint changes.

Figure 4.7: Graph shows matching results from 50 image pairs using SIFT keypoints, predefined keypoints, and after further optimization. The matching results are further improved after eliminating correspondence pairs that have different complex pattern responses.

Figure 4.8: 2nd data set: Graph shows matching results from 50 image pairs using SIFT keypoints, predefined keypoints, and after further optimization. The matching results are further improved after eliminating correspondence pairs that have different complex pattern responses.

4.5 Chapter Summary

In this chapter, we have combined the local invariant features with the more traditional corner-like features for selecting salient keypoints used for matching in urban environments. These keypoints are generally more useful for matching than the original SIFT keypoints since they are validated to comprise one or more common corner-like patterns that often exist in man-made objects, e.g., buildings and windows. This also reduces the vast number of SIFT keypoints that are not all equally important. However, our work and experiments are only targeting urban scenes which often comprise several of these corner-like patterns, e.g., proper corner and T-junction. Thus, it may not work on other objects, e.g., an electric fan.

After matching keypoints using the Euclidean distance based nearest neighbor approach, we further verify the matches by scrutinizing the difference in the complex pattern responses of each correspondence pair. Those correspondence pairs with dissimilar pattern responses are eliminated as they are mainly false matches (outliers). From our experimental results, we perceive the effectiveness of the approach where the matching results have been highly optimized. However, since the patterns of the image patches will be deformed with respect to the viewpoint changes, this verification approach cannot handle large viewpoint changes very well, e.g., for view angles that separate images higher than 40° .

In the next two chapters (Chapters 5 and 6), we introduce approaches that are more robust to large viewpoint changes.

Chapter 5

Matching of Building Facades under Large Viewpoint Change

Chapter Overview. We present a novel approach to locate a large list of point correspondences in urban scenes despite the presence of repetitive structures and widely separated views. We extract hypotheses of building facades based on a robust line fitting algorithm. Each hypothesis is defined by a planar convex quadrilateral in the image, which we call a “q-region”. A projective transformation model can be derived from each pair of q-regions in the two images. If the pair of q-regions is correctly aligned, all line segments and interest points within them will fit the transformation model and match accordingly. Consequently, the largest pair of correctly aligned q-regions will produce the largest list of matches. Extensive experimental results demonstrate the robustness of our approach in which we achieved a tenfold increase in true matches for facades separated by wide angles when compared to state-of-the-art approaches such as SIFT and MSER. We also show that using our approach results in increased accuracy in image retrieval.

5.1 Introduction

Wide-baseline matching in urban scenes is one of the most challenging tasks in computer vision. Due to the highly repetitive nature of the man-made environments, traditional

wide-baseline matching approaches are rendered ineffective [75, 132]. Feature descriptors such as SIFT (Scale Invariant Feature Transform) [87] (Section 2.3.1) that are based on region patches are even more problematic under large viewing angle variations. In the literature, several appearance-based object recognition techniques have been proposed, where most of them include a feature detector searching for salient features (e.g., points or regions) characterized by corners, edges, or entropy, and followed by a feature descriptor to describe the features. To try obtaining invariance under arbitrary viewing conditions, they either extract invariant features or they compute invariant descriptors based on non-invariant features. Nevertheless, none of the existing approaches (e.g., by Baumberg [10]; Bay et al. [12]; Brown and Lowe [21]; Kadir et al. [61]; Lee et al. [77]; Lowe [89]; Matas et al. [92]; Mikolajczyk and Schmid [97]; Obdržálek and Matas [116]; Pritchett and Zisserman [118]; Tell and Carlson [148, 149]; Tuytelaars and Van-Gool [156]; Xiao and Shah [172]; etc.) can handle large viewpoint changes very well, e.g., for view angles that separate images higher than 40° [98]. The features corresponding over two wide baseline frames cannot be effectively determined via traditional matching techniques due to the large geometric transformation changes [172]. The problem of obtaining a large list of true correspondences in urban scenes across widely separated views is yet to be solved.

In this chapter, we introduce a novel approach [74] that is dedicated to the matching of building facades in urban scenes despite their ambiguity in scales, unit patches, and skewness of the matching patterns. We hypothesize quadrilateral regions of building facades in a pair of images. The fundamental idea behind our approach is that if two regions are true corresponding pair of regions between the two images, they will fit a projective transformation model [75]. Similarly, all line segments and interest points within the pair of regions will also fit this projective transformation model. As a result, the largest pair of correctly aligned regions will comprise the largest list of correspondences, in terms of

line segments and interest points within the pair of regions. The main contributions of this approach are: (1) in the region extraction stage, we use a line fitting algorithm to locate all possible planar convex quadrilaterals that are often used in man-made objects, e.g., buildings. (2) in the matching stage, we use a projective transformation model to locate the best pair of correctly aligned regions. This pair of regions comprises the largest list of line and point correspondences. In contrast with existing matching approaches or any appearance-based matching approaches, our matching approach can consistently establish a large list of point correspondences irrespective of the change in the viewpoint and the existence of multiple repetitive and symmetric structures.

5.2 Related Work

Detecting and matching features across two images typically involves a feature detector and a feature descriptor, which we have previously discussed in Chapter 2. Among these feature detectors and feature descriptors, many are proposed to be invariant to rotation, affine transformation, and/or scale in order to account for variations in the appearance due to the change of viewpoint. They can be combined in various ways to suit different systems.

To recapitulate, the Harris corner detector [54] (Section 2.2.2) is probably the most widely used feature detector, which is based on the eigenvalues of the second-moment matrix. Although it is called a corner detector, it is not selecting just corners, but rather any image location that has large gradients in all directions at a predetermined scale. It produces a high number of detections and is repeatable. However, this detector is not invariant to scale and affine transformations. The Harris-Laplace and Hessian-Laplace detectors [95] (Section 2.2.3) were then proposed to deal with affine transformations. They use a (scale-adapted) Harris measure or the determinant of the Hessian matrix to select the location, and the Laplacian to select the scale. To improve on speed, Lowe [87]

approximates the Laplacian of Gaussian (LoG) by a Difference of Gaussians (DoG) filter (Section 2.2.5). However, the drawback of the LoG and DoG detectors is that local maxima can also be detected in the neighborhood of contours or straight edges, where the signal change is only in one direction. These maxima are less stable because their localization is more sensitive to noise or small changes in neighboring texture.

A technique proposed by Baumberg [10] extracts Harris points at multi-scales, and then adapts the shape of the region to the local image structure using an iterative procedure based on the second moment matrix [85]. The Harris points as well as the associated regions are not invariant to large affine transformation changes since they are not extracted in an affine invariant way. Furthermore, there are numerous features repeated at the neighboring scale levels which introduces higher complexity and more false matches. Schaffalitzky and Zisserman [131] extend the Harris-Laplace detector [95] by affine normalization proposed by Baumberg [10]. However, the location and scale of features are still provided by the scale invariant Harris-Laplace detector [95] which is not invariant to large affine transformations.

Tuytelaars and Van-Gool [155] detect salient features in an affine invariant way by using nearby edges of Harris points. Using two nearby edges, which are required for each feature, limits the number of potential features in an image. A parallelogram region is bounded by these two edges and the initial Harris point. Several intensity based functions are needed to determine the parallelogram and a reliable approach is needed to extract the edges. In their later work [156], the method is purely intensity-based and starts with the extraction of local intensity extrema. The algorithm then investigates the intensity profiles along rays going out of the local extremum, and finally an ellipse is fitted to the region determined by significant changes in the intensity profiles. A similar approach based on local intensity extrema was introduced by Matas et al. [92], named the Maximally Stable Extremal Regions (MSER) detector (Section 2.2.6). They

use the water-shed algorithm to find intensity regions and fit an ellipse to the estimated boundaries. The MSER detector, in particular, has been evaluated in [98] to have better performance than other affine invariant detectors.

For feature descriptors, Mikolajczyk and Schmid [99] had evaluated a variety of them and had identified the Scale invariant feature transform (SIFT) [87] (Section 2.3.1) and SIFT-based descriptors to be the most resistant to common image deformations and achieving the best performance. In Lowe’s original scheme, he uses the DoG detector [87] to detect features which are invariant to scale changes. In a recent evaluation [105], the SIFT descriptor is found to be the best descriptor for the MSER detector. The MSER detector with the SIFT descriptor has been used recently in a large scale object recognition system by Nistér and Stewénus [114].

Other similar descriptors such as Gradient Location and Orientation Histogram (GLOH) [99] (Section 2.3.4), Principle Components Analysis-SIFT (PCA-SIFT) [66] (Section 2.3.2), Speeded Up Robust Features (SURF) [12] (Section 2.3.5), etc., are proposed after the SIFT descriptor either as an extension or an evolution of it. These descriptors are invariant to orientation and illumination changes just like the way the SIFT descriptor is. They were also shown to have similar accuracy with the SIFT descriptor in feature matching [99], but are usually more compact (lower dimension) leading to improvements in computation time. The local descriptors work well only when the region patches have distinct appearance characterized by either color or texture. In cases where the perspective fore-shortening effects become dominant, the affine models are no longer appropriate.

In our experiments in this chapter, we compare our matching results with the following approaches: Harris-Affine detector with SIFT descriptor; MSER detector with SIFT descriptor; and the DoG detector with SIFT descriptor.

Putting the detectors and descriptors aside, there are various papers targeting man-made environments. Several of them ([30, 68, 102, 120, 140]) have taken advantages of

geometric lines for providing geometric hints about the camera orientation parameters and focal length in urban environments. A popular example is the estimation of orthogonal vanishing directions. However, the assumption that the majority of detected line segments comes from principal vanishing directions associated with the world coordinate frame has to be valid; despite other problems such as noise and lens imperfections. In contrast with our approach, we do not detect any vanishing lines and points.

Coorg and Teller [30] propose a system that utilizes vanishing points to determine the orientation of dominant building facades relative to a panoramic camera. However, a combination of specialized hardware (e.g., GPS and inertial sensors) is needed. Robertson and Cipolla [120] describe a system that uses line segments to recover dominant vanishing points in images. The images are transformed into a canonical frame corresponding to a rectified view of the dominant plane. Image features are then detected, followed by pose recovery between views. However, the system is unable to distinguish between some viewpoints without additional information, e.g., extra query views, due to the similarity of buildings or parts of buildings. Moreover, the triangulation using known GPS locations has not been carried out. Similarly, Stamos and Allen [140] use vanishing points for pose determination. Their system requires a detailed geometric model obtained by laser range finding and they have only localized views in the vicinity of a single building.

Košecká and Zhang [68] present an approach in which each rectangular structure is defined by four line segments coming from two different orthogonal line's groups. A multi-stage matching strategy is used to match these rectangular structures, e.g., using the ratios of rectangular sizes and normalized cross correlation of canonical views. Given that these ratios and appearances change significantly under large viewpoint changes, their matching approach may hence be unreliable. In addition, their approach can neither deal with general symmetries, nor with the inevitable ambiguity in pose recovery results.

Recently, Wang et al. [164] introduce Line Signature which its construction is based on curves. The curves are extracted from images and are approximated with line segments.

A line signature is a local cluster of the line segments. The approach is good in handling low-texture images and large viewpoint changes of non-planar scenes.

5.3 Methodology

Our approach is based on the observation that the facades of many structures in man-made environments comprise some planar quadrilaterals, e.g., buildings, windows, etc. Each quadrilateral does not have to have four right angles nor does it have to have four equal sides, as the shape of, e.g. a square, will look different under different viewing angles. While these quadrilaterals may be easily perceived by humans, automatic detecting of these quadrilaterals from images is not straightforward. We start by finding the edges. Each quadrilateral is defined by four edges, and in images these edges (of man-made objects) can be effectively characterized by line segments [111]. For each of the four line segments of a quadrilateral, there exists two adjacent corners. Each corner is formed by the intersection of two line segments, which we call a “geometric corner”. Geometric corners are different from appearance-based corners (e.g., those detected by Harris corner detector [54] (Section 2.2.2)) as geometric corners are always real corners in man-made objects where each corner comes from an intersection of two line segments.

As illustrated in Figure 5.1, we can locate and hypothesize a quadrilateral in a building facade by first detecting a geometric corner, followed by detecting line segments that can intersect with the two extended line segments from the geometric corner to form a quadrilateral. We call this planar quadrilateral a “q-region” [75]. This q-region is a region within the image and is formed from line segments instead of corners. It corresponds closely to a planar structure in the real world so that we can derive a projective transformation (homography) model from each respective pair of q-regions in two images. The line segments from the geometric corner are always extended to the boundary of the

image for robustness towards occlusions (e.g., by trees) and robustness against broken line segments.

Figure 5.1: Hypothesizing a quadrilateral of a building facade. The quadrilateral can be perceived by first locating an initial corner (in the highlighted area), and then linking to other edges to form a quadrilateral.

Our approach is robust since the outline of a q-region comprises not just one, but four corners. The assumption is that at least one of the four geometric corners can be detected in order to get the outline of a building facade like the one shown in Figure 5.1. In cases where any or all four geometric corners are undetected due to a non-quadrilateral shape in the outline of the building facade, or due to occlusion, fragmentation or missing edge pixels, there are other geometric corners within the building facade that can be detected, e.g., from windows, doors, etc. Thus quadrilaterals can still be perceived. An example is shown in Figure 5.2.

As we have no prior knowledge to which are the corresponding q-regions that exist in both images, we have to extract all possible q-regions that could occur in both images. This can be achieved by taking all possible combinations of the detected edge segments to form q-regions, and subsequently eliminating the false ones.

Figure 5.2: A quadrilateral is perceived from other corners and edges within the building facade.

A projective transformation (homography) model can be derived from each pair of q-regions in the two images. Extracted line segments and Harris points within each pair of q-regions are used to validate the projective transformation model. If the pair of q-regions is correctly aligned, all the line segments and Harris points will fit the projective transformation model and will match accordingly. The largest pair of correctly aligned q-regions will produce the largest list of matches.

The presented approach eradicates unnecessary parts of the scene in one image which does not exist in the other image by constricting the pair of images to the q-regions enclosing the same scene. This reduces the chance of obtaining false matches in scenes comprising repetitive and symmetric patterns (e.g., building facades). It differs from many existing approaches which blindly match the entire image and inevitably obtain a significant number of false matches. In general, our algorithm involves three key stages as follows:

- (i) Extract and fix broken line segments (Section 5.3.1).
- (ii) Hypothesize possible q-regions in the images (Section 5.3.2).
- (iii) Align q-regions to find the pair with highest confidence (Section 5.3.3).

In the following sections, we discuss each of these three stages.

5.3.1 Line fitting algorithm

We extract line segments from an edge image by a strip fitting algorithm [81]. Let $L = \{\ell_1, \dots, \ell_n\}$ represents a set of line segments found from [81]. Due to the well-known fragmentation of edge maps of real images, missing edge pixels will result in the breaking of a line segment. To circumvent this problem, a post-processing step is applied to identify and fix broken line segments. Two line segments ℓ_i and ℓ_j are joined if the following condition is satisfied,

$$\begin{aligned} & Cond_{join}(\ell_i, \ell_j) : \\ & \theta > \tau_\theta \text{ and } \min(\|P_{k,i}, P_{k,j}\|_2) < \tau_d, \forall k \in \{1, 2\}, \end{aligned} \quad (\text{Eq. 5.1})$$

where θ is the smaller internal angle between ℓ_i and ℓ_j , and τ_θ denotes the allowed internal angle threshold between ℓ_i and ℓ_j . $P_{k,i}$ indicates the k^{th} end point of ℓ_i , and τ_d denotes the allowed distance threshold between the shortest end-to-end point of ℓ_i and ℓ_j . We denote $\ell_{i,j}$ as the line segment formed by joining ℓ_i with ℓ_j , and $\hat{L}$ to be the set of new line segments that are created. The complete set of line segments that will be used in the next stage is $\{L \cup \hat{L}\}$, where all line segments before and after the post-processing step are retained. Thus there is no risk of joining two line segments that are not supposed to be joined (not broken) and end up losing them permanently. However, line segments that are relatively short (e.g., < 5 pixels) in $\hat{L}$ are removed as they are mainly insignificant lines or noise.

5.3.2 Q-region hypothesis extraction

The outputs from the line fitting approach in the previous stage are used to locate possible q-regions in the images. The approach is presented below.

Figure 5.3: Illustration of q-region formation. (a) Two line segments ℓ_i and ℓ_j detected. (b) Initial region $R_{i,j}$ (shaded green) formed by an intersection point $(x_{i,j}, y_{i,j})$, and two line segments ℓ_i and ℓ_j . (c) Refined region $R_{i,j,k}$ formed by two intersection points $(x_{i,j}, y_{i,j})$ and $(x_{i,k}, y_{i,k})$, and three line segments ℓ_i , ℓ_j , and ℓ_k . (d) Q-region $R_{i,j,k,l}$ formed by four intersection points $(x_{i,j}, y_{i,j})$, $(x_{i,k}, y_{i,k})$, $(x_{j,l}, y_{j,l})$, and $(x_{k,l}, y_{k,l})$, and four line segments ℓ_i , ℓ_j , ℓ_k , and ℓ_l .

5.3.2.1 Initial geometric corner detection

We hypothesize a q-region by first detecting geometric corners. As discussed, for a planar quadrilateral with four edges and corners, our assumption is that one out of the four corners can be detected in order to get the entire quadrilateral as a q-region. We detect geometric corners, each described as an intersection point between two line segments. Given two line segments ℓ_i and ℓ_j (see Figure 5.3(a)) as first and second line segments

with their end-to-end points as P_1P_2 and P_3P_4 respectively, $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $P_3 = (x_3, y_3)$, and $P_4 = (x_4, y_4)$, the intersection point $(x_{i,j}, y_{i,j})$ between ℓ_i and ℓ_j in parametric form is defined as

$$x_{i,j} = x_1 + t_i(x_2 - x_1), \quad (\text{Eq. 5.2})$$

$$y_{i,j} = y_1 + t_i(y_2 - y_1). \quad (\text{Eq. 5.3})$$

Solving for t_i yields

$$t_i = \frac{(x_4 - x_3)(y_1 - y_3) - (y_4 - y_3)(x_1 - x_3)}{(y_4 - y_3)(x_2 - x_1) - (x_4 - x_3)(y_2 - y_1)}. \quad (\text{Eq. 5.4})$$

If the denominator is not 0, then ℓ_i and ℓ_j are not parallel and are therefore either intersected or intersecting. We define a geometric corner as

$$\min(\|(x_{i,j}, y_{i,j}), P_{k,i}\|_2) < \tau_d, \quad (\text{Eq. 5.5})$$

$$\text{and } \min(\|(x_{i,j}, y_{i,j}), P_{k,j}\|_2) < \tau_d, \forall k \in \{1, 2\}, \quad (\text{Eq. 5.6})$$

where τ_d denotes the allowed distance threshold between the intersection point $(x_{i,j}, y_{i,j})$ and the two closest end points from ℓ_i and ℓ_j . This approach is not exhaustive since not every two line segments can intersect within τ_d . The point $(x_{i,j}, y_{i,j})$ does not have to be within the image because a corresponding geometric corner may be shifted out of view in the second image. An initial region $R_{i,j}$ detected at this stage is formed by the intersection point $(x_{i,j}, y_{i,j})$, together with the the first and second line segments ℓ_i and ℓ_j extended to the boundary of the image as shown in Figure 5.3(b).

5.3.2.2 Q-regions formation

We have to find from each geometric corner all the other supporting line segments to form q-regions. Let ℓ_k with end-to-end points P_5P_6 be the line we are going to check if it can be a possible third line segment to form a q-region. We hypothesize a possible

third line segment for a q-region if at least one end point P_5 or P_6 is within $R_{i,j}$, and $y_{i,k} < y_{i,j}$ where $(x_{i,k}, y_{i,k})$ is the point where ℓ_i and ℓ_k intersect. The reason why we detect at least one end point instead of both end points to be within $R_{i,j}$ is to deal with cases where one end point is outside $R_{i,j}$ as shown in Figure 5.4(a). Figure 5.3(c) shows

Figure 5.4: Other atypical cases of q-region formation. (a) Case where not both end points P_5 and P_6 are within $R_{i,j}$ (P_5 is outside and to the left of $R_{i,j}$). (b) Case where the bottom-left corner of the building facade (front face) is outside the image. Intersection point $(x_{i,k}, y_{i,k})$ can still be established.

a third line segment is found and we have our refined region as $R_{i,j,k}$. The intersection point can be achieved by applying the same equations as Eq. 5.2 - Eq. 5.4. Similarly, $(x_{i,k}, y_{i,k})$ does not have to be within the image. This is for dealing with building facades that are partially shifted out of the image as shown in Figure 5.4(b).

Finally, we find the fourth line segment in a similar fashion as the third line segment ℓ_k found previously. Let ℓ_l with end-to-end points P_7P_8 be the line we are going to check if it can be a possible fourth line segment to form a q-region. However instead of finding only one point where ℓ_i and ℓ_k intersect, now we find two intersection points, one between ℓ_k and ℓ_l and the other between ℓ_j and ℓ_l . We hypothesize a possible fourth line segment for a q-region if ℓ_l is already intersected with ℓ_j and ℓ_k , or at least one of it's

end point P_7 or P_8 is within $R_{i,j,k}$. In addition, $x_{i,j} < x_{j,l}$ where $(x_{j,l}, y_{j,l})$ is the point where ℓ_j and ℓ_l intersect; and $x_{i,k} < x_{k,l}$ where $(x_{k,l}, y_{k,l})$ is the point where ℓ_k and ℓ_l intersect. Figure 5.3(d) shows the formation of a q-region $R_{i,j,k,l}$ which the entire front face of the building facade is found. This process is repeated to find all possible third and fourth line segments for generating possible q-regions using this geometric corner. Small (insignificant) q-regions with $area < \tau_{area}$ will be removed. In our work, we denote τ_{area} as $\frac{1}{9} \times area\ of\ image$ which has shown to be very stable throughout our experiments.

5.3.2.3 Other geometric corners of the facade

The example discussed assumes a geometric corner is initially found on the top-left corner of a building facade. However, the initial geometric corner can also be found on the bottom-left, top-right, or bottom-right corner of a building facade. The steps remain very similar and the resulting q-region will still be the same. For example, if the initial geometric corner is found on the bottom-left corner of a building facade, the initial region $R_{i,k}$ and the final resulting q-region $R_{i,j,k,l}$ formed will be as shown in Figure 5.5(a) and 5.5(b) respectively. This is a duplicated q-region and it will be removed in a later step even though it was generated from a different initial geometric corner.

5.3.2.4 Possible combinations of q-regions

In order to better deal with the issue that part(s) of the building facade may be occluded or shifted out of the image, we generate all possible combinations of q-regions using all the vertical and horizontal line segments detected, not necessarily to use the original four line segments. Figure 5.6 illustrates an example where the building facade is partially shifted out of the image and there are four vertical line segments including the first line segment ℓ_i detected. The number of q-regions formed is given by the combination formula

$${}_nC_r = \frac{n!}{r!(n-r)!} = \frac{n(n-1)(n-2)\dots(n-r+1)}{r!}, \quad (\text{Eq. 5.7})$$

Figure 5.5: Finding q-region with a different initial geometric corner on another side of facade (bottom-left corner). (a) Initial region $R_{i,k}$ (shaded green) formed by an intersection point $(x_{i,k}, y_{i,k})$, and two line segments ℓ_i and ℓ_k . (b) Resulting q-region $R_{i,j,k,l}$.

where n is the number of line segments from which we can choose, r is the number of line segments to be chosen at one time, and $n!$ denotes the factorial of n . To reduce computational load, q-regions with $area < \tau_{area}$ will be removed.

5.3.3 Q-regions alignment

As each q-region has four corner points $(x_{i,j}, y_{i,j})$, $(x_{i,k}, y_{i,k})$, $(x_{j,l}, y_{j,l})$, $(x_{k,l}, y_{k,l})$, as illustrated previously in Figure 5.3, we take the respective four corner points from each pair of q-regions in the two images to instantiate a projective transformation (homography) model (see Figure 5.7). The homography equation is given by

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \quad (\text{Eq. 5.8})$$

If the pair of q-regions is correctly aligned (true corresponding pair), the points within the pair of q-regions should be corresponded. This is homography matching as previously discussed in Section 2.4.3. To achieve rotation invariance, we can reorder the sequence (e.g., in a clockwise manner) of the corner points for each q-region.

Figure 5.6: All possible combinations of q-regions (shaded green) formed from 4 vertical line segments ℓ_1 , ℓ_2 , ℓ_3 , and ℓ_i , giving ${}_4C_2 = 6$ q-regions.

Let P_k and P'_k denote a point in the first and second image respectively. We represent the homography model as $\mathbf{H}_m$ and denote the projection of point P_k by $\mathbf{H}_m$ to be $\mathbf{H}_m(P_k)$ as illustrated in Figure 5.7. We use the previously extracted line segments

Figure 5.7: A projective transformation (homography) model is instantiated by four corresponding corner points in a pair of q-regions, and a point P_k from one q-region (on left) is projected to another q-region (on right) as $\mathbf{H}_m(P_k)$.

within each pair of q-regions to evaluate the homography model. To supplement the evaluation accuracy and the finding of more salient corresponding points, we also exploit Harris points [54] (Section 2.2.2). In our previous work [75], it has been shown that the Harris corner points extracted from the building facades are very useful for evaluating the correctness of the homography model. The correctness of the homography model and hence the confidence between a pair of q-regions is then computed as

$$Conf(\mathbf{H}_m) = \sum_k \left[\|\mathbf{H}_m(P_k), P'_k\|_2 \leq \tau_d \right] + \sum_{\ell_n} \sum_{P_j \in \ell_n} \left[\|\mathbf{H}_m(P_j), P'_j\|_2 \leq \tau_d \right], \quad (\text{Eq. 5.9})$$

where

$$P'_* = \underset{i}{\operatorname{argmin}} \left(\|\mathbf{H}_m(P), P'_i\|_2 \right), \quad (\text{Eq. 5.10})$$

τ_d is the threshold distance offset, and $[\cdot]$ is a binary indicator function. As a result, the largest pair of correctly aligned q-regions will obtain the highest confidence. Figure 5.8 displays the results of the corresponding line segments and Harris points within a pair of q-regions with the highest confidence. These corresponding Harris points, which are

more salient features, are returned as a list of point correspondences between the two images. The summary of our algorithm is presented in Algorithm 5.1 below.

Algorithm 5.1

Input: Two images of urban scene.

- I. Extract line segments from the two images.
- II. Identify and join broken line segments.
- III. Locate geometric corners.
- IV. For each geometric corner:
 - (i) Extend the two line segments that form the geometric corner.
 - (ii) Find all possible third and fourth line segments that can intersect with the two extended line segments to form quadrilateral region which we call a q-region.
- V. For each q-region:
 - (i) Find all possible combinations of q-regions using all vertical and horizontal line segments detected within the q-region.
- VI. Remove any duplicated and small ($area < \tau_{area}$) q-regions.
- VII. For each q-region in one image:
 - (i) Derive a homography model with each q-region ($\times 4$ for rotation invariance) in another image.
 - (ii) Evaluate the homography model with line segments and Harris points within the pair of q-regions.
- VIII. Select the pair of q-regions with highest confidence.

Output: A list of point correspondences.

Figure 5.8: Corresponding line segments and Harris points within a pair of correctly aligned q-regions. Areas that are not in the q-regions are obscured. (a) 8225 pixels of corresponding line segments in total. (b) 134 corresponding Harris points in total.

5.4 Experimental Setup

In the following, we first describe the three other approaches that will be used for result comparisons. Next, we discuss the evaluation criteria used, and finally the image data sets used in the experiments. We run two experiments. The first experiment examines each approach’s robustness to changes in scale, orientation, and viewpoint. The second experiment evaluates the approaches on image retrieval. The results will be presented in Section 5.5.

5.4.1 Other approaches for comparison

Three of the most popular current approaches as discussed in Chapter 2 are used in the comparisons with our approach as follows:

- (i) Harris-Affine detector [96, 97] with SIFT descriptor [89] (Sections 2.2.4 and 2.3.1)
- (ii) MSER detector [92] with SIFT descriptor [89] (Sections 2.2.6 and 2.3.1)
- (iii) DoG detector [87] with SIFT descriptor [89] (Sections 2.2.5 and 2.3.1)

In order to decide on acceptance or rejection of a candidate match for each of the three approaches indicated above, we need a metric in appearance space. We use the distance ratio introduced by Lowe [89]. This measure compares the ratio of the distances in the appearance of the query point to its best and its second best matches. The criterion used is the ratio of these two distances defined as

$$\frac{d(P_a, P_{a1})}{d(P_a, P_{a2})}, \quad (\text{Eq. 5.11})$$

where P_a , P_{a1} and P_{a2} denote the query point, its best and its second best matches respectively. This ratio characterizes how distinctive a given query point is, and avoids ambiguous matches. A low value implies that the best match performs significantly better than its best contender, and is thus considered a reliable match. A high value implies that the points are not distinctive enough relatively to each other, and thus the match is to be rejected to avoid a false match.

We exploit RANSAC homography [38, 55] (homography estimation using RANSAC) to improve on the above three approaches by rejecting false matches. RANSAC (random sample consensus) [41] is a robust estimation procedure that uses a minimal set of randomly sampled point correspondences to estimate image transformation parameters, and finds a solution that has the best consensus with the data. We select sets of $r = 4$ point correspondences and compute the homography $\mathbf{H}_m$ between them using the direct linear transformation (DLT) method [55]. We repeat this with $n = 500$ trials and select the solution that returns the largest number of inliers, whose projections are consistent with $\mathbf{H}_m$ within a tolerance of ϵ pixels. Given the probability that a candidate match is correct between an image pair (the inlier probability) is p_i , the probability of finding the correct transformation after n trials is

$$p(\mathbf{H}_m \text{ is correct}) = 1 - (1 - (p_i)^r)^n. \quad (\text{Eq. 5.12})$$

After a large number of trials the probability of finding the correct $\mathbf{H}_m$ is very high. For example, for an inlier probability $p_i = 0.5$, the probability that the correct $\mathbf{H}_m$ is not found after 500 trials is approximately 1×10^{-14} .

We have performed separate experiments with and without RANSAC homography and find that RANSAC homography can improve on the above three approaches by rejecting many false matches. It has been proven to work very well in many applications (e.g., in [22, 89, 132, 156, 177]) and will only fail when the number of false matches is too severe, e.g., much above 50%. The results discussed in our experiments will include only the ones with RANSAC homography for the three approaches for a fairer comparison.

5.4.2 Evaluation criteria

Our evaluation criteria are different for evaluating the approaches on the two experiments, i.e. matching tasks under various transformations and image retrieval.

Our evaluation criteria for matching tasks under various transformations is mainly based on the number of true matches and the number of false matches obtained for an image pair. A true match is where the pair of matched points comes from the same physical location, and a false match is where the pair of matched points comes from two different physical locations. In addition, for cases like matched points on the leaves of a tree, we do not consider them as true matches. We visually inspect the correctness of each match as we cannot use homography to inspect the correctness of the matches because our approach is based on homography.

We perform experiments for three types of transformations: scale, orientation, and viewpoint. The results are presented by the average matching ratio of n image pairs versus the change in scale/orientation/viewpoint, and the average number of true matches in the n image pairs versus the change in scale/orientation/viewpoint. The number n is the number of image pairs used in the evaluation, and the matching ratio is the number of

true matches relative to the total number of matches computed as

$$\text{Matching ratio} = \frac{\# \text{ true matches}}{\# \text{ true matches} + \# \text{ false matches}}. \quad (\text{Eq. 5.13})$$

Only a single match is allowed for each point. Intuitively, we want the matching ratio to be high, as close to 1 as possible.

Our evaluation criterion for image retrieval is rather straightforward, in which for a given query image, each approach will attempt to retrieve the corresponding image in the database. The corresponding image should be the one with the largest number of matches among all images in the database. A score is only awarded if the corresponding image is returned in the top three positions. If the corresponding image is returned in the first position, the approach is awarded 3 points; if the corresponding image is returned in the second position, it is awarded 2 points; for third position, it will be awarded 1 point; otherwise, it is awarded no points. The maximum score for 40 queries will be $40 \times 3 = 120$ points. The total score for each approach is then divided by 120 to give the percentage of correct match scores.

5.4.3 Data sets

We use two different data sets for our two experiments. The first data set is used to evaluate the four approaches on images of urban scenes with various geometric transformations. Figure 5.9 and Figure 5.10 show examples from the data set used for the evaluation. Three different changes in imaging conditions are evaluated: Scale changes (Figure 5.10(a)); rotation (Figure 5.10(b)); and viewpoint changes (Figure 5.10(c)).

The scale change sequence is acquired by varying the camera zoom, for which the scale factor varies from 1.4 to 2.5. Image rotations are obtained by rotating the camera around its optical axis in the range of 10 to 45 degrees. In the case of viewpoint change sequence, the camera position varies from a fronto-parallel view to one with foreshortening from

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 5.9: Data set. Examples of images used for the evaluation: (a)-(i) An image example from each of the 9 sets of images used for the evaluation of four approaches on matching under various transformations.

approximately 15 to above 50 degrees. Each of these 3 sequences contains 6 images with a gradual geometric transformation and therefore there are 18 images in a set. We acquired 9 sets representing different urban scenes, and thus there are 162 images in total. An image example from each of the 9 sets of images is shown in Figure 5.9. All images are of medium resolution (640×480 pixels).

(a)

(b)

(c)

Figure 5.10: Data set. Examples of images used for the evaluation: (a) Zoom (scale changes), (b) rotation (orientation changes), (c) viewpoint changes. In the experimental comparisons, the left most image of each set is used as the reference image.

The second data set is used in the image retrieval. For this experiment, we select a data set of 40 image pairs from the ZuBud database [135] which were acquired by two

different cameras. Each image pair comprises urban scene photographed from different viewpoints. There are also some occlusions by trees and other objects in some images. All images are of medium resolution (640×480 pixels). We separate the image pairs into two databases each with 40 images, one used as a query database and the other as a reference database. Each image in the query database will be used as a query into the reference database to find its corresponding image, one at a time.

5.5 Experimental Results

This section provides results for the comparisons between our approach and the other three approaches on the geometric transformation and image retrieval experiments. The two experimental results are presented in Section 5.5.1 and Section 5.5.2 respectively.

5.5.1 Matching under various transformations

In the following, we present results for the performance for scale changes, orientation changes, and viewpoint changes.

5.5.1.1 Scale changes

Figure 5.11 plots results of the experiment where the main image transformation is a scale change. Figure 5.11(a) shows the average matching ratio and Figure 5.11(b) shows the average number of true matches. Harris-Affine + SIFT + RANSAC homography approach performs slightly better for most of the test scales. This was expected due to the automatic scale selection applied in Harris-Affine detector. However, given that most of the approaches also display an average matching ratio of above 0.8, this is not very significant. MSER + SIFT + RANSAC homography approach is more sensitive to scale change where the average number of true matches dropped below 100 when the scale change reached a factor of 2 and above. For scale changes, the results of our approach are comparable to the best scores in this experiment.

Figure 5.11: Evaluation for scale changes of a factor 1.4-2.5. (a) Average matching ratio. (b) Average number of true matches.

Figure 5.12: Evaluation for orientation changes of 10-45 degrees. (a) Average matching ratio. (b) Average number of true matches.

5.5.1.2 Orientation changes

In this experiment, all the result curves are rather horizontal and they are at similar average matching ratio values as shown in Figure 5.12(a), i.e. all approaches have a similar performance (about ± 0.1 differences). This is apparent as these approaches are known to be rotation invariant. Looking closely, we can see that the average number of true matches as displayed in Figure 5.12(b) for MSER + SIFT + RANSAC homography approach is lower than the rest; however, its average matching ratio is still above 0.8. On the other hand, our approach has a slightly higher average number of true matches for all the test angles, and average matching ratio is comparable with the rest with no significant differences.

5.5.1.3 Viewpoint changes

In this experiment, we evaluate the performance for viewpoint changes to above 50 degrees. This introduces a perspective transformation which can locally be approximated by an affine transformation. Mikolajczyk and Schmid [99] pointed out that this is one of the most challenging transformations. Now we can see a significant difference in the results on average matching ratio and average number of true matches for the approaches as displayed in Figure 5.13(a) and Figure 5.13(b) respectively. Our approach is dramatically better at handling large viewpoint changes where the average matching ratio and average number of true matches remain stable even after reaching the $> 50^\circ$ viewpoint angle limit. In contrast, the results for the rest of the approaches decrease tremendously when the viewpoint angle gets higher; barely any true matches can be found when the $> 50^\circ$ viewpoint angle limit is reached. For viewpoint angle of about 35° and above, we achieved a tenfold increase in true matches when compared to the other three approaches. This shows the robustness of our approach in viewpoint changes for building facades.

Figure 5.14 displays a matching example for images with a viewpoint change of more than 50 degrees on the example images in Figure 5.10(c). Matching these images is

Figure 5.13: Evaluation for viewpoint changes of 15 to > 50 degrees. (a) Average matching ratio. (b) Average number of true matches.

extremely challenging due to the large viewpoint difference and the repetitive patterns in the scene. To our knowledge, we have not come across any existing approaches that can effectively match images like these. There are barely any true matches obtained from the other three approaches (see Figure 5.14(a) - 5.14(c)). Our approach clearly dominates them by obtaining over hundreds of true matches with a matching ratio of 0.91 (see Figure 5.14(d)).

5.5.2 Image retrieval

We have also evaluated the four approaches using an image retrieval experiment for urban scenes [135] taken from different viewpoints. The setup, evaluation criterion, and data set are as discussed in Section 5.4. The scores are given in Table 5.1. The results show that our approach's matching accuracy translates into better retrieval results.

Table 5.2 presents a detailed look at one of the image retrieval examples. The potential matches are presented in rank order for each approach. None of the approaches except our approach could rank the correct image in the first position. DoG + SIFT + RANSAC

Table 5.1: Percentage of images correctly retrieved in the image retrieval experiment.

Approach	Correct retrieval
Harris-Affine + SIFT + RANSAC homography	65%
MSER + SIFT + RANSAC homography	57%
DoG + SIFT + RANSAC homography	82%
Our approach	90%

Table 5.2: An image retrieval example. The top three matches (Rank 1-3) for a query image for the four approaches are shown. “RANSAC Hm” denotes RANSAC homography.

Query image		Correct image to retrieve
Rank	1		2		3	
Harris-Affine + SIFT + RANSAC Hm # matches		12		9		8
MSER + SIFT + RANSAC Hm # matches		11		10		10
DoG + SIFT + RANSAC Hm # matches		13		12		10
Our approach # matches		134		47		31

(a) Harris-Affine + SIFT + RANSAC homography

(b) MSER + SIFT + RANSAC homography

(c) DoG + SIFT + RANSAC homography

(d) Our approach

Figure 5.14: A matching example for images of building facades (from Figure 5.10) with a viewpoint change of $> 50^\circ$. Four approaches for comparison (a-d). (a) Harris-Affine detector + SIFT descriptor + RANSAC homography: 7 matches, 0 pair of true match, matching ratio 0. (b) MSER detector + SIFT descriptor + RANSAC homography: 8 matches, 0 pair of true match, matching ratio 0. (c) DoG detector + SIFT descriptor + RANSAC homography: 6 matches, 0 pair of true match, matching ratio 0. (d) Our approach: 260 matches, 237 pairs are true matches, matching ratio 0.91.

homography approach managed to rank the correct image in the second position, while Harris-Affine + SIFT + RANSAC homography approach ranked it in the third position. MSER + SIFT + RANSAC homography approach did not manage to get the correct image falling into any of the top three positions.

Figure 5.15 shows the image matching results for the query image and the correct image from Table 5.2. Our approach clearly dominates the other three approaches with

matches in the order of 100s. DoG + SIFT + RANSAC homography approach obtained slightly above 10 matches for this pair of images which successfully ranked the correct image in the second position. The rest of the approaches could barely hit 10 matches for this pair of images. However, with 8 matches, Harris-Affine + SIFT + RANSAC homography approach just managed to seize the correct image in the third position.

(a) Harris-Affine + SIFT + RANSAC homography

(b) MSER + SIFT + RANSAC homography

(c) DoG + SIFT + RANSAC homography

(d) Our approach

Figure 5.15: Matching results for the query image and the corresponding image from Table 5.2. Four approaches for comparison (a-d). (a) Harris-Affine detector + SIFT descriptor + RANSAC homography: 8 matches, 3 pairs are true matches. (b) MSER detector + SIFT descriptor + RANSAC homography: 9 matches, 1 pair is true match. (c) DoG detector + SIFT descriptor + RANSAC homography: 12 matches, 10 pairs are true matches. (d) Our approach: 134 matches, 130 pairs are true matches.

5.6 Chapter Summary

In this chapter, we have presented a novel concept for locating a substantial number of point correspondences between images of urban scenes despite the presence of multiple repetitive structures and large viewing variations. The proposed approach first extracts line segments using a robust line fitting algorithm. Next, we find quadrilaterals in the images using these line segments since man-made objects (e.g. building facades) often comprise some planar convex quadrilaterals. We named the detected quadrilaterals as “q-regions”. The fundamental concept behind our approach is that if two q-regions are correctly aligned in two images, they will fit a projective transformation (homography) model. As a result, all detected line segments and Harris points within the pair of q-regions will also fit this transformation model and match accordingly. We locate the largest pair of correctly aligned q-regions, which produces the largest list of correspondences.

We compared our matching performance with other popular state-of-the-art approaches in experiments on scale changes, orientation changes, and viewpoint changes. There were no significant differences between the approaches in the performance on scale and orientation changes. Nevertheless, for viewpoint changes, the comparison has shown that the performance of others declined with similar rates as the change of viewpoint angle increases, and only our approach remained stable throughout with a tenfold more true matches as compared to the others. We also demonstrated our approach in an image retrieval experiment, where the results favored our approach for its increased accuracy. These experiments revealed the strength and robustness of our approach.

In contrast to our previous approaches in Chapter 3 and Chapter 4 which eliminate false correspondences (outliers), our approach in this chapter increases the number of true correspondences and is robust to large viewpoint changes. Nevertheless, this approach targets urban environments and may not work well in images which do not comprise

any man-made structures, e.g., images of forest scenes. In occasions when two images have very dense detected corner points (e.g., from trees or/and multi-window building facades), a number of false correspondences may still occur (by homography matching). We will address this limitation in the next chapter.

Chapter 6

PRIUS: Projective Region Invariants for Urban Scenes

Chapter Overview. In this chapter, we extend our approach in Chapter 5 to a fully affine invariant descriptor, coined PRIUS (Projective Region Invariants for Urban Scenes). Interest points and their neighboring points within close spatial proximity are used to describe the detected quadrilateral regions, and to robustly align building facades through their projectivity. Improvements upon our previous approach include 70% more true matches; and robustness to the asymmetric problem in corner detection, false corresponding regions and noise in images. Experimental results show that using the PRIUS descriptor can achieve about twenty-fold more true matches than SIFT and Affine-SIFT. We also demonstrate the applicability of PRIUS to a mobile guide application.

6.1 Introduction

Image matching aims at finding correspondences between two images of the same scene, which is a fundamental step in many computer vision applications. 3D reconstruction [129], image registration [158, 173], object recognition [17, 87], and robot localization [18, 89, 114] are just a few examples. The extracted image features can often be represented in different ways before matching them. For example, an edge can be

represented as a boolean variable at each image point that describes whether an edge is present or not at that point. This feature representation is referred to as a (feature) descriptor. Some feature descriptors have been previously discussed in Section 2.3. Several prior work on descriptors invariant to rotation, affine transformations and/or scale have been proposed in [89, 92, 96, 130, 156], while many recent ones [12, 66, 99, 107] are introduced after SIFT (Scale Invariant Feature Transform) [89] (Section 2.3.1) either as an extension or an evolution of it. However, none of them can robustly deal with images of urban scenes which have shown to be much more complex [68, 75, 132]. The high degree of symmetry and repetition in these man-made environments induces a significant number of false correspondences; while a substantial dissimilarity in viewpoints can hamper the number of correspondences to a modest number. Work on descriptors that have been proposed to solve the above-mentioned problems are limited, and to the best of our knowledge we have not come across any effective ones.

In this chapter, we propose a region descriptor which in combination with our matching scheme is invariant to large viewpoint changes for building facades. The approach is an extension to our previous work [74, 75] (Chapter 5) which defines a projective transformation model between hypothesized quadrilateral regions of building facades, and uses interest points to validate them. In contrast with our previous approach, we associate a descriptor called PRIUS (Projective Region Invariants for Urban Scenes) to each quadrilateral region. Each descriptor is a collection of feature vectors, where each vector comprises of an interest point and its neighboring points based on spatial proximity. These descriptors are matched by their projectivity to locate the best pair of corresponding regions, which is the one with the largest number of corresponding points in the two images. By building a relationship between each interest point and its neighboring points, our approach can robustly deal with two common problems. (1) Asymmetric numbers of interest points detected in the two images, since an interest point detected

in one image may not appear in the other image and therefore results in a lesser number of corresponding points. These missing (undetected) interest points may never be recovered. (2) False corresponding points found in the two images after projectivity due to noise, e.g., a tree in one of the images will comprise a massive number of corner points which can be easily mismatched after projectivity.

While our approach can increase the number of corresponding points, recover missing (undetected) interest points, and provide better robustness to noise, it is also invariant to image translation, scaling, rotation, and affine projection.

6.2 Related Work

A large variety of feature descriptors have been proposed in the literature, e.g., Gaussian derivatives [42], moment invariants [100], complex features [10, 131], steerable filters [46], phase-based local features [26], and Scale Invariant Feature Transform (SIFT) [89] (Section 2.3.1). The latter one proposed by Lowe has been shown to be the most resistant to common image deformations and outperformed the others [99]. Recently, other detectors such as the Maximally Stable Extremal Regions (MSER) [92] (Section 2.2.6) has also been used with the SIFT descriptor in a large scale object recognition system by Nistér and Stewénus [114]. However, experiments [75] show that the Difference of Gaussians (DoG) [89] (Section 2.2.5) achieves better matching results than MSER in images of urban scenes when combined with the SIFT descriptor. Various similar descriptors as previously discussed in Section 2.3, and the most recent one Affine-SIFT (ASIFT) [107] (Section 2.3.6), are proposed after SIFT and claim to be more robust, distinctive, compact, and/or invariant. However, all these local descriptors work well only when the region patches have distinct appearance characterized by either color or texture. In cases where the perspective foreshortening effects become dominant, the affine models are no

longer appropriate. We compare our matching results with DoG + SIFT, and ASIFT in our experiment in Section 6.4.

The more recent work targeting man-made environments is proposed by Schindler et al. [132] which attempts to overcome the symmetry and repetition problems in images of urban environments. It detects multiple perspectively distorted periodic 2D patterns in an image and matching them to a 3D database of textured facades by reasoning about the underlying canonical forms of each pattern. However, the method performs well only when several tall buildings that exhibit highly repetitive structures in the images exist.

Our previous approach [74, 75] (Chapter 5) first extracts hypotheses of building facades, each defined by a planar convex quadrilateral in the image. It follows by finding the pair of corresponding quadrilateral regions in the two images with the largest number of correspondences after a projective transformation. Although the approach works well under large viewing variations, it may fail due to noise (e.g., trees in the image) and may miss (undetected) interest points. Furthermore, the approach is difficult to be applied to other applications (e.g., reconstruction of 3D scenes) as it is not even in a descriptor form. We will therefore formalize our approach as a descriptor and will compare matching results with our previous approach in experiments described in Section 6.4.

6.3 Methodology

We use the quadrilateral region extraction technique [74, 75] proposed in our previous chapter to locate quadrilateral regions as it has shown promising results. We propose a novel region descriptor called PRIUS (Projective Region Invariants for Urban Scenes) that describes each of the extracted quadrilateral regions by the detected interest points, and their neighboring points based on spatial proximity within the region. We do not form any feature vectors based on pixel intensities which are sensitive to geometric distortion, e.g., due to viewpoint change. Subsequently, the descriptors in the two images

are matched by the projectivity of the saved interest points. The pair of descriptors with the largest list of corresponding points indicates a match between the two regions. We detect interest points using the FAST-ER corner detector which is recently proposed by Rosten et al. [124]. By using some machine learning techniques to derive the detector, it has outperformed other existing detectors (e.g., Harris [54] and DoG [87]) in terms of speed and repeatability.

In cooperation with our previous techniques of extracting quadrilateral regions and corners, our descriptor is poised to produce promising results.

6.3.1 PRIUS descriptor and matching

The proposed descriptor does not contain any pixel intensity information as the change in camera position will induce a deformation of the object, and any appearance-based information may subsequently be invalid. For every hypothesized quadrilateral region, we describe each detected FAST-ER corner point [124] and its nearest 12 neighboring points as one vector. Hence, a PRIUS descriptor has $N \times 13$ vectors for N number of FAST-ER corner points in a quadrilateral region. We only have to match these descriptors in different images. Let us consider a 3D coordinate frame and two quadrilateral regions of the same scene but with different camera angles. Let's call the two quadrilateral regions R_1 and R_2 as shown in Figure 6.1. R_2 is defined by the point $\vec{b}_0$ and two linearly independent vectors $\vec{b}_1$ and $\vec{b}_2$ contained in the region. Let us consider a FAST-ER corner point $\vec{P}_2$ in R_2 . Since the vectors $\vec{b}_1$ and $\vec{b}_2$ form a basis in R_2 , we can express $\vec{P}_2$ as

$$q_1 \vec{b}_1 + q_2 \vec{b}_2 + \vec{b}_0 = \begin{pmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ 1 \end{pmatrix} = B \vec{q}, \quad (\text{Eq. 6.1})$$

where $B = \begin{pmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_0 \end{pmatrix} \in \mathbb{R}^{3 \times 3}$ defines the quadrilateral region R_2 , and $\vec{q} = (q_1, q_2, 1)^T$ defines the 2D coordinates of $\vec{P}_2$ with respect to the basis $(\vec{b}_1, \vec{b}_2)$. We can compute a

similar identity for quadrilateral region R_1 as

$$\vec{P}_1 = A \vec{s}, \quad (\text{Eq. 6.2})$$

where $A = (\vec{a}_1, \vec{a}_2, \vec{a}_0) \in \mathbb{R}^{3 \times 3}$ defines R_1 , and $\vec{s} = (s_1, s_2, 1)^T$ defines the 2D coordinates of $\vec{P}_1$ with respect to the basis $(\vec{a}_1, \vec{a}_2)$. We impose the constraint that point $\vec{P}_1$ maps to point $\vec{P}_2$ under perspective projection centered at the origin, giving

$$\vec{P}_1 = \alpha(\vec{q}) \vec{P}_2, \quad (\text{Eq. 6.3})$$

where $\alpha(\vec{q})$ is a scalar that depends on $\vec{P}_2$, and consequently on $\vec{q}$. By combining the equation above with the constraint that $\vec{P}_1$ and $\vec{P}_2$ must be situated in its corresponding region, we obtain the relationship between the 2D coordinates of these points given by

$$\vec{s} = \alpha(\vec{q}) A^{-1} B \vec{q}, \quad (\text{Eq. 6.4})$$

where the role of $\alpha(\vec{q})$ is to simply scale the term $\alpha(\vec{q}) A^{-1} B \vec{q}$ such that its third coordinate is 1. We can represent $\alpha(\vec{q}) A^{-1} B$ as a homography matrix $\mathbf{H}_m$ (Section 2.4.3) and compute the above equation as

$$\vec{s} = \mathbf{H}_m \vec{q}. \quad (\text{Eq. 6.5})$$

If R_1 and R_2 are true corresponding regions, the FAST-ER corner point $\vec{P}_2$ and its 6 nearest neighboring points (shaded blue in Figure 6.1) in R_2 should fit the homography matrix $\mathbf{H}_m$ to correctly locate the 7 corresponding FAST-ER corner points in R_1 . This is homography matching as previously discussed in Section 2.4.3. There are more than 6 nearest neighbors (total of 12) stored in a descriptor vector although only 6 are used. This is to solve the asymmetric problem where a point can be detected in one quadrilateral region but not in the other, and thus the nearest 6 neighbors may be slightly different in this case. E.g., the 6th nearest neighbor for $\vec{P}_2$ in R_2 may be the 7th nearest neighbor

Figure 6.1: Point projectivity between two quadrilateral regions R_1 and R_2 of the same scene but different camera angles.

for $\vec{P}_1$ in R_1 . Therefore, we store slightly more than 6 nearest neighbors to overcome this problem.

Observations after various experiments summarized the point-to-point homography matching results into different cases. Given two extracted quadrilateral regions (may be true or false corresponding pair) R_3 and R_4 , a FAST-ER corner point (let's call it $\vec{P}_3$), its 6 nearest neighboring points $\{\vec{P}_{3a}, \vec{P}_{3b}, \dots, \vec{P}_{3f}\}$, and their corresponding points in R_4 (may be true or false corresponding points) denoted as $\vec{P}_4, \{\vec{P}_{4a}, \vec{P}_{4b}, \dots, \vec{P}_{4f}\}$, the 7 frequent cases after projectivity of the 7 points by $\mathbf{H}_m$ from R_3 to R_4 can be categorized as:

- (i) All 7 points $\vec{P}_4, \vec{P}_{4a}$ to $\vec{P}_{4f}$ are found.
- (ii) $\vec{P}_4$ is found, and most neighboring points (≥ 4) are found.
- (iii) $\vec{P}_4$ is not found, but most neighboring points (≥ 5) are found.
- (iv) $\vec{P}_4$ is found, and few neighboring points (< 4) are found.
- (v) $\vec{P}_4$ is not found, and few neighboring points (< 5) are found.

(vi) $\vec{P}_4$ is found, but no neighboring point (< 1) is found.

(vii) All 7 points $\vec{P}_4, \vec{P}_{4a}$ to $\vec{P}_{4f}$ are not found.

We determine from experiments that the candidate points falling within cases (i)-(iii) give a very high accuracy of correct matches. As a result, we only consider a candidate point as valid if it falls within these three cases. This approach is very robust since it does not evaluate the pair of candidate points in the two quadrilateral regions individually, but as a whole with the neighbors in order to adjudicate if the pair is a correct match. Consequently, corner points coming from other objects or structures, e.g., the dense corner points detected on a tree, will have very low chances of being matched.

We are also the first to recover interest points that are not detected in one image but in the other. This can be easily understood from case (iii) where $\vec{P}_4$ is not found but most of the neighboring points (at least 5) are found. We can recover such missing (undetected) interest point in an image if the point exists in the other image and most of their neighboring points are corresponded. This solves the common problem in corner detection where the number of detected corner points in two images are asymmetric. Given that the number of corner points are asymmetric in the pair of quadrilateral regions, in this case R_3 and R_4 , we can apply an inverse homography matrix $\mathbf{H}_m^{-1}$ to the unmatched corner points in R_4 to locate any corresponding corner points in R_3 in the same manner as before.

Eventually, the best pair of corresponding quadrilateral regions (correctly aligned) in the two images will obtain the largest list of corresponding FAST-ER corner points. The summary of our algorithm is presented in Algorithm 6.1 below.

Algorithm 6.1

Input: Two images of urban scene.

- I. Extract line segments from the two images.
- II. Identify and join broken line segments.
- III. Locate geometric corners.
- IV. For each geometric corner:
 - (i) Extend the two line segments that form the geometric corner.
 - (ii) Find all possible third and fourth line segments that can intersect with the two extended line segments to form quadrilateral region which we call a q-region.
- V. For each q-region:
 - (i) Find all possible combinations of q-regions using all vertical and horizontal line segments detected within the q-region.
- VI. Remove any duplicated and small ($area < \tau_{area}$) q-regions.
- VII. For each q-region in one image:
 - (i) Detect interest points using FAST-ER corner detector.
 - (ii) Describe each detected corner point and its nearest 12 neighboring points as one vector.
- VIII. For each descriptor in one image:
 - (i) Match with each descriptor in another image by their projectivity of the saved corner points in each vector.
 - (ii) The point-to-point homography matching results into 7 possible cases. Only three of them are considered valid cases. Some corner points that are not detected in one image but in the other image can be recovered, depending on which cases they fall in.
- IX. Select the pair of q-regions with the largest list of corresponding corner points.

Output: A list of point correspondences.

6.4 Experimental Results

We compare our PRIUS image matching performance with three of the approaches that have previously shown superior results in image matching.

- (i) **SIFT (with DoG detector)** [89] (Sections 2.2.5 and 2.3.1): The most popular approach and has previously shown to outperform other approaches, e.g., MSER [92] and Harris-Affine [96] in matching images of urban scenes under large viewpoint changes [75].
- (ii) **Affine-SIFT** [107] (Section 2.3.6): Very recent state-of-the-art approach and has shown to be much more robust to viewpoint changes than other state-of-the-art approaches. Found more matches than MSER, SIFT, Harris-Affine, etc. when the viewpoint change is substantial.
- (iii) **Lee et al.’s approach** [74, 75] (Chapter 5): Our previous approach which is dedicated to matching building facades in urban scenes. Has previously shown to outperform state-of-the-art approaches, e.g., SIFT and MSER in matching images of urban scenes under large viewpoint changes. Found 10 times more matches than the other approaches in the experiments.

We perform RANSAC homography [38, 55] on SIFT and Affine-SIFT to reject false matches as it has been proven to work very well in many applications (e.g., in [22, 89, 156]). We plot only the number of true matches in the experiments since the number of false matches is very small. The number of false matches is small because all the approaches are either homography based or are exploiting RANSAC homography. In order to make the experiments as fair as possible, we have also implemented FAST-ER corner detector [124] in Lee et al.’s approach for detecting corner features. Results have shown improvement to the originally used Harris detector [54] in the paper.

(f)

Figure 6.2: Image matching sequence results of four approaches. Image sequence is shown in (a)-(e). Matching results of different approaches are shown in (f). “ab” denotes the matching of image sequence (a) with image sequence (b).

We select 50 images from the ZuBud database [135] which comprises views of building scenes where each building has a sequence of 5 images taken from different viewpoints and/or scales as shown in Figure 6.2. By matching the first image in the sequence (e.g., Figure 6.2(a)) with the rest of the images in the sequence, we can plot the matching results of the different approaches in Figure 6.2(f). We see in Figure 6.2(f) that the matching goes well for all approaches in the first two image matching sequences (“ab” and “ac”), where the number of true matches hit 50 and above for all. However, when the viewpoint of the buildings gets wider (e.g., in “ad” and “ae”), SIFT and Affine-SIFT can barely reach above 10 matches. Lee et al.’s approach is still able to hold on to the

100 matches line in image matching sequence “ae”. On the other hand, PRIUS has over 70% more matches than Lee et al.’s approach, and almost twenty-fold more matches than SIFT and Affine-SIFT in the last image matching sequence “ae”. The matching results of the approaches for the image matching sequence “ae” are displayed in Figure 6.3.

(a) SIFT + RANSAC homography

(b) Affine-SIFT + RANSAC homography

(c) Lee et al.’s approach

(d) PRIUS

Figure 6.3: Matching results of four approaches for image matching sequence “ae” in Figure 6.2. SIFT+RANSAC homography: 11 total matches, 8 true matches. Affine-SIFT + RANSAC homography: 9 total matches, 9 true matches. Lee et al.’s approach: 107 total matches, 100 true matches. PRIUS: 180 total matches, 176 true matches. In (c) and (d), we emphasize the corresponding quadrilateral regions in the two images by obscuring the areas that are not parts of the regions.

We perform the same experiment to a total of 50 images from the ZuBud database and we plot the average matching results in Figure 6.4. We can see that the results are quite consistent with the ones in Figure 6.2. We see a significant difference between

PRIUS and the other approaches on the average number of true matches, especially when the viewpoint change is large. In any of the image matching sequences, PRIUS has more true matches than all the other approaches, and its average number of true matches remains stable throughout the image matching sequences. PRIUS is dramatically better at handling large viewpoint changes in urban environments than SIFT and Affine-SIFT. The average matching results again reveal that PRIUS has over 70% more matches than Lee et al.’s approach, and almost twenty-fold more matches than SIFT and Affine-SIFT when the viewpoint difference is large (see Figure 6.4). The experimental results have shown that PRIUS has outperformed the three approaches in matching images of urban scenes.

Figure 6.4: Average matching results of the four approaches over 50 images.

Other than the effectiveness in image matching, we also evaluate the robustness of the approaches in dealing with false corresponding quadrilateral regions. For a pair of false corresponding regions, the number of matches should be as low as possible. This allows the approach to be more effective in selecting the best corresponding regions in the two images while robust to outliers (false matches) and noise such as corner points from trees. We compare Lee et al.’s approach to PRIUS and the image results are

displayed in Figure 6.5. The results show that PRIUS is significantly more robust to false corresponding regions than Lee et al.’s approach, where only few false matches are obtained by PRIUS in the experiment. On the contrary, Lee et al.’s approach has obtained close to 100 false matches. This is predictable as PRIUS does not only evaluate each pair of candidate points in the two quadrilateral regions, but also on each of their neighboring points based on spatial proximity.

(a) Lee et al.’s approach: 92 false matches.

(b) PRIUS: 14 false matches.

Figure 6.5: Matching results of a pair of false corresponding quadrilateral regions in two images. The pair of false corresponding quadrilateral regions is specified by obscuring the areas that are not parts of the regions. The image results of Lee et al.’s approach and PRIUS are shown in (a) and (b) respectively. Less number of matches indicates better robustness to outliers (false matches) in wrong regions.

6.5 Mobile Guide Application

The work presented in this section is motivated by a conceptual mobile guide application. The idea is that by using a mobile phone with a built-in camera, the user can transmit a photograph image of his surrounding to a central computer system. Given the approximate knowledge of the handset’s position obtained by the Global Positioning System (GPS), the system can compare the query view with the closest reference view in the database in order to determine pose. Location information (e.g., library, cafeteria) can then be transmitted back to the user and overlaid onto the image.

However, there are several challenges that need to be addressed in order to integrate the individual components to the system and making it work in reality. One of them is the formulation of a database for cost-effective data storage and efficient retrieval of the reference views. Another notable challenge is on the wide-baseline matching stage, e.g., in dealing with the highly repetitive nature of the man-made environments and the variations of the viewing angle.

In this section, we describe a mobile guide system [76], and integrate our approach PRIUS to the system to perform wide-baseline matching between views. An evaluation of the system is done in our system evaluation section.

6.5.1 System Overview

In contrast with the robot localization system described by Se et al. [134], we determine the poses of the query views by referring to a much simpler model comprising a database of views of scenes. Figure 6.6 illustrates the process flow of the entire system. Using

Figure 6.6: The process flow of the mobile guide system.

a camera phone, the user can transmit a photographic image of his surrounding (real world scene) with its GPS position through the wireless network to the system server. The system compares the query view with the closest reference view in the database in order to determine pose. Location information are then transmitted back to the user and overlaid onto the image.

6.5.2 Database formulation

Our image database as shown in Figure 6.6 comprises panoramic views of our North Academic Complex in Nanyang Technological University (Singapore), as well as a number of other buildings of interest outside the campus. All views are tagged by their GPS locations. Each panoramic view is formed by stitching multiple photograph images covering 360° view at each position. Each position interval is approximately 10-25 meters, depending on the space availability. Each panoramic view diminishes overlapping between individual views (e.g., 36 images covering 360° orientation) and thus reduces storage overhead. Moreover, since each database view is in a panoramic form covering all orientation at a position, there is no concern about which direction (e.g., North, East, NorthEast, etc.) the query view is taken from.

For efficiency, we compute the descriptions of the database views offline (in advance). Only a description of the panoramic view with the closest GPS position with the query view is required for wide-baseline matching.

6.5.3 Pose recovery

Regardless of which wide-baseline matching technique is used, at least four correspondences between the two images are needed to perform pose recovery between views from planar homographies. The homography equation is defined as

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \quad (\text{Eq. 6.6})$$

Information about the scene can subsequently be transmitted back to the user and overlaid onto the query image. Some image examples are shown in Figure 6.7.

Figure 6.7: Examples of location information overlaid onto the query images.

6.5.4 System Evaluation

This experiment evaluates the proposed mobile guide system, as well as PRIUS which is the core of the system. The setup for PRIUS used for wide-baseline matching between views is as described in Section 6.3. We revisited the area where the photographs of the database images were captured, at different times of day, and acquired 95 images as query views. The query views were taken with different resolution camera phones, and some of them contained clutter and noise, e.g., pedestrians, traffic, trees, etc.

Using the framework described earlier, we attempt to determine the pose of each of the 95 query views and overlay the location information onto the query images. However, instead of representing these results with labeled information on the query images, we represent them with their pose determination results by sketching corresponding building facade outlines in the database views and query views. This can be achieved by using the recovered homographies as described in Section 6.5.3, and this method gives a good indication of the accuracy in pose estimations. The representative results for some query examples are shown in Figure 6.8.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.8: Illustrative of some recovered database views. (a) shows a panoramic view in the database with the nearest GPS position with query views (b)-(f). (b)-(f) show the query views. The corresponding building facade outlines demonstrate the accuracy of the recovered pose estimations. In (d), there is no recovered view from the panoramic database image because there are no corresponding regions being detected between the two views.

Overall, 90 out of the 95 queries are registered correctly with the correct corresponding views recovered from the panoramic database images. When the system fails, it is mainly because the query images are too dim as some of these photograph images were taken in the evening. They are difficult to recognize even for humans. An image example of this is shown in Figure 6.8(d). Another reason is because the image scene does not comprise any building facades. Our system fundamentally detects man-made structures such as building facades. However, this assumption is not too restrictive in our mobile guide application since the users can easily be instructed to point the camera at some buildings of interest.

6.6 Chapter Summary

In this chapter, we have proposed a novel design of a region descriptor called PRIUS (Projective Region Invariants for Urban Scenes), and have highlighted its potential in matching images of urban scenes across different views. Many of the state-of-the-art approaches are rendered ineffective in dealing with these scenes due to the highly repetitive nature of the man-made environments and the variations of the viewing angle. We incorporate the recent approaches of quadrilateral region extraction (our previous approach [74, 75] discussed in Chapter 5) and corner detection (FAST-ER corner detector [124] discussed in Section 2.2.7), together with our proposed descriptor PRIUS in this chapter to tackle these problems.

Our approach can reliably locate a large list of point correspondences in urban scenes despite the presence of repetitive structures and widely separated views. In addition, it is also robust to the asymmetric problem in corner detection, false corresponding regions and noise in images. We first extract hypotheses of building facades defined by planar quadrilateral regions in the images and describe them as PRIUS descriptors. The description is based on the positions of extracted FAST-ER corner points and their few nearest neighboring points in each region. The descriptors are matched through their projectivity. Missing (undetected) corners in one image are recovered. The pair of aligned quadrilateral regions with the largest list of corresponding FAST-ER corner points is therefore attained.

We have attempted to compare matching results with some of the state-of-the-art approaches which have previously shown good results in matching objects with dissimilar viewpoints, e.g., SIFT and the recently proposed Affine-SIFT (ASIFT). The experimental results indicate that PRIUS has about twenty-fold more true matches than these approaches when the viewpoint difference is large. It has also outperformed our previous

approach in [74, 75] (Chapter 5), achieving over 70% more true matches and showing better robustness to outliers.

We have also integrated PRIUS to a mobile guide system to perform wide-baseline matching. An evaluation was done on the proposed mobile guide application with a database of panoramic views of scenes tagged by GPS locations. Only the view with the closest GPS position with the query view is acquired for wide-baseline matching. Above 90% of the queries were registered correctly with the correct pose determinations. These experiments revealed the strength and robustness of our approach and system. A limitation of the system is that the user has to be instructed to point the camera at some buildings of interest and not objects of interest. The system will fail if the scene comprises only non-planar object such as a human statue, or even a bicycle.

Chapter 7

Conclusions

7.1 Summary and Discussion of Results

The opening chapter of this thesis was devoted to specifying the problems and research goals of this thesis. This work has primarily focused on the following problems associated with wide-baseline matching in urban environments: high degree of symmetry and repetitive structures results in outliers (false matches); large viewpoint variations and occlusions result in limited matches (correspondences). In Chapter 2, we explored through background studies of the existing methods, typically involving the three key processes - feature extraction, feature description, and feature matching.

In Chapter 3, we introduced a verification model that can be used to enhance image matching results by eliminating outliers. We exploited several geometric and appearance-based measurements, and conditional probabilities were used to compute the probability of each true correspondence. Experimental results revealed that the approach can eliminate all or most outliers. A comparison with some commonly used methods (e.g., RANSAC) showed that our verification model was less sensitive to large percentage of outliers ($\#outliers > \#inliers$), and thus produced better results. However, our verification model is only eliminating/reducing outliers, it does not increase the number of true correspondences. As shown in our experimental section, the number of resulting correspondences were limited when the viewing angle variation was large.

In Chapter 4, we provided an alternative to hypothesize and verify salient keypoints and true correspondences in urban scene images using an image pattern verification technique. We combined the local invariant features with the more traditional corner-like features for selecting salient keypoints used for matching images of urban environments. These keypoints are generally more useful for matching than the original SIFT keypoints since they are validated to comprise one or more common corner-like patterns that often exist in man-made environments. We further verified the matches by eliminating correspondences that had dissimilar pattern responses which were most likely false matches. While this method can improve matching performance by pre-selecting salient keypoints before matching and removing outliers after the match, it cannot increase the number of true correspondences or deal with large viewing angle variations.

In order to better deal with large viewing angle variations, we proposed an approach in Chapter 5 which extracts hypotheses of building facades each defined by a planar quadrilateral region in the image. This quadrilateral region was named “q-region”. The approach finds the pair of corresponding q-regions in the two images with the largest number of correspondences after a projective transformation (homography). The idea is that if a pair of q-regions is correctly aligned, all line segments and interest points within them will fit the transformation model and match accordingly. Consequently, the largest pair of correctly aligned q-regions will produce the largest list of matches.

We compared our matching performance with other popular state-of-the-art approaches (e.g., SIFT and MSER) in experiments on scale changes, orientation changes, and viewpoint changes. There were no significant differences between the approaches in the performance on scale and orientation changes. However for viewpoint changes, the comparison had shown that the performance of others declined with similar rates as the change of viewpoint angle increased, and only our approach had remained stable throughout with tenfold more true matches compared to the others. The approaches were also evaluated

in an image retrieval experiment, where the results favored our approach for its increased accuracy (90% correct retrieval).

Although the approach works well under large viewing variations, it is not robust to noise (e.g., false corners from trees in the image). The dense detected corner points on the trees may result in many false correspondences. Furthermore, the approach is difficult to be applied to other applications (e.g., reconstruction of 3D scenes) as it is not even in a descriptor form.

In Chapter 6, we extended our approach (in Chapter 5) to a fully affine invariant descriptor, coined PRIUS (Projective Region Invariants for Urban Scenes). We associated a PRIUS descriptor with each q-region which was extracted as in Chapter 5. The descriptor was based on the positions of extracted FAST-ER corner points and their neighboring points within close spatial proximity in each q-region. We did not form any feature vectors based on pixel intensities as they were sensitive to geometric distortion, e.g., due to viewpoint change. The descriptors were matched by their projectivity to locate the best pair of corresponding regions, which were the ones with the largest number of corresponding points in the two images. This approach is very robust since it does not evaluate the pair of candidate points in the two q-regions individually, but as a whole with the neighbors in order to adjudicate if the pair is a correct match. Consequently, corner points coming from other objects or structures, e.g., the dense corner points detected on a tree, will have very low chances of being matched. We were also able to recover missing (undetected) interest points that were not detected in one image but in the other image using our approach.

Improvements upon our previous approach in Chapter 5 include 70% more true matches; and robustness to the asymmetric problem in corner detection, false corresponding regions and noise in images. Experimental results had also showed that using the PRIUS descriptor can achieve about twenty-fold more true matches than state-of-the-art approaches such as SIFT and Affine-SIFT.

We have also demonstrated the applicability of our approach (PRIUS) to a mobile guide application. The application allowed the user to find information of his surrounding in the outdoor environments using a camera-equipped mobile device. We constructed a database of panoramic views by photographing in our campus as well as a number of other buildings of interest outside the campus, classified by their GPS coordinates. We registered each query image to the closest model in the database. PRIUS was used to match between the views and location information such as places of interest were overlaid on the query image after pose recovery (using homographies). Above 90% of the queries were registered correctly with the correct pose determinations. These experiments revealed the strength and robustness of our approach and system. The failures were mainly due to the reason that the scenes did not comprise any buildings or quadrilateral objects.

The experiments reported in this thesis were based on images of urban scenes, comprising mainly views of building facades. Matching of these scenes across different views is the central theme of this thesis. We have primarily looked into aspects such as corner-like feature patterns, textures, corner points, quadrilateral man-made structures, etc. Thus, our approaches may not work well in other categories of images, e.g., images of object.

7.2 Future Work

Incorporating color. One direction of research for the future would be to incorporate color. Most computer vision applications tend to ignore color information, where color information is usually discarded and only luminance information is considered. The color of an image only find its use in obtaining the luminance as we can see from the low-level applications (e.g., corner detector) to high-level applications (e.g., object recognition). There are many examples from nature where color is used by animals and plants to

entice or warn the others [146]. The manufacturing sector also uses color extensively in packaging to market goods.

Color is neglected as a recognition cue as many variations can cause significant changes in measured color. Nevertheless, some techniques [49, 160] are introduced to reduce the sensitivity of color information to photometric changes and make color description more robust. Weijer et al. [159, 160] extend the SIFT local feature descriptor with color information by concatenating a color descriptor to it with a fixed weighting. They have shown that their combination of color and shape outperformed a pure shape-based approach.

Incorporating color allows the feature descriptor to distinguish several differently colored objects with the same shape. Colored textures such as wood, marble and metal, are hard to recognize without color. One main limitation in incorporating with color is that it will increase the computational complexity.

Dealing with non-planar regions. A major drawback of our approaches is the implicit assumption of the planarity of the scene, whereby there exists buildings that are made up of non-planar regions. We will have to look into plausible ways to deal with these non-planar regions, e.g., by partially tessellating the non-planar scene into piece-wise patches and match these patches without using homography-based method. Or we could estimate the epipolar geometry from apparent contours of smooth curved surfaces with affine camera models.

Detecting 2nd best region pair. In this thesis, we did not detect the *2nd* best corresponding pair of non-overlapping regions to avoid complications. However, this is possible, and some image results obtained by PRIUS are shown in Figure 7.1 below.

(a) A pair of 3D house images.

(b) Best pair of corresponding regions.

(c) 2nd best pair of corresponding regions.

Figure 7.1: Image results of two pairs of corresponding regions detected in a pair of 3D house images from the Visual Geometry Group, Oxford University [161]. This pair of images is known for its matching difficulty due to the large viewpoint difference and repetitive structures. The pair of original images is shown in (a). The best pair of corresponding regions obtained by PRIUS is shown in (b), with 104 matches. The 2nd best corresponding pair of non-overlapping regions obtained by PRIUS is shown in (c), with 25 matches. There are altogether 129 matches obtained. The corresponding regions in (b) and (c) are specified by obscuring the areas that are not parts of the regions.

References

- [1] A. Aberneithy. *Automatic Detection of Calcified Nodules of Patients with Tuberculous*. University College London, UK, 2007.
- [2] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, pages 883–885. Dover Publications, 9th printing, New York, 1972.
- [3] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. on Graphics (SIGGRAPH)*, 25(3):853–861, July 2006.
- [4] M. M. Ajmal. *Managing knowledge in project-based organizations: A cultural perspective*, page 154. University of Vaasa, Finland, Nov 2009.
- [5] M. S. Aksoy, O. Torkul, and I. H. Cedimoglu. An industrial visual inspection system that uses inductive learning. *Journal of Intelligent Manufacturing*, 15(4):569–574, Aug 2004.
- [6] G. Arfken. *Eigenvectors, Eigenvalues*, pages 229–237. Academic Press, Orlando, FL, 3rd edition, 1985.
- [7] G. Arfken. *Taylor’s Expansion*, pages 303–313. Academic Press, Orlando, FL, 3rd edition, 1985.
- [8] F. Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183–193, May 1954.
- [9] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition (PR)*, 13(2):111–122, April 1981.

- [10] A. Baumberg. Reliable feature matching across widely separated views. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 774–781, June 2000.
- [11] H. Bay, A. Ess, T. Tuytelaars, and L. Van-Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, June 2008.
- [12] H. Bay, B. Fasel, and L. Van-Gool. Interactive museum guide: Fast and robust recognition of museum objects. In *Proc. 1st Int'l Workshop on Mobile Vision (IMV), held in conjunction with 9th European Conference on Computer Vision (ECCV)*, May 2006.
- [13] H. Bay, T. Tuytelaars, and L. Van-Gool. SURF: Speeded up robust features. In *Proc. 9th European Conference on Computer Vision (ECCV)*, volume 1, pages 404–417, May 2006.
- [14] S. Belongie and J. Malik. Matching with shape contexts. In *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL)*, page 20, June 2000.
- [15] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 831–837, Nov 2000.
- [16] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. 8th IEEE Int'l Conference on Computer Vision (ICCV)*, volume 1, pages 454–461, July 2001.
- [17] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(4):509–522, April 2002.
- [18] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke. Metric localization with scale-invariant visual features using a single perspective camera. In *Proc. 1st European Robotics Symposium (EUROS)*, pages 195–209, March 2006.

- [19] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 26–33, June 2005.
- [20] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(4):712–727, April 2008.
- [21] M. Brown and D. G. Lowe. Invariant features from interest point groups. In *Proc. 13th British Machine Vision Conference (BMVC)*, pages 656–665, Sept 2002.
- [22] M. Brown and D. G. Lowe. Recognising panoramas. In *Proc. 9th IEEE Int'l Conference on Computer Vision (ICCV)*, volume 2, pages 1218–1225, Oct 2003.
- [23] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int'l Journal of Computer Vision (IJCV)*, 74(1):59–73, Aug 2007.
- [24] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. J. Wiley and Sons, May 2009.
- [25] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur. *A Theory of Shape Identification*. Lecture Notes in Mathematics. Springer Verlag, 2008.
- [26] G. Carneiro and A. D. Jepson. Multi-scale phase-based local features. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 736–743, June 2003.
- [27] P. C. Cattin, H. Bay, L. Van-Gool, and G. Székely. Retina mosaicing using local features. In *Proc. 9th Int'l Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 185–192, Oct 2006.
- [28] W.-C. Chen and P. Rockett. Bayesian labelling of corners using a grey-level corner image model. In *Proc. 4th IEEE Int'l Conference on Image Processing (ICIP)*, volume 1, pages 687–690, Oct 1997.

- [29] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 44–51, June 2000.
- [30] S. Coorg and S. Teller. Automatic extraction of textured vertical facades from pose imagery. Technical Report 729, Massachusetts Institute of Technology, Computer Graphics Group, Cambridge, MA, Jan 1998.
- [31] A. Criminisi, I. D. Reid, and A. Zisserman. A plane measuring device. *Image and Vision Computing (IVC)*, 17(8):625–634, June 1999.
- [32] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *Int'l Journal of Computer Vision (IJCV)*, 10(2):101–124, April 1993.
- [33] P. G. T. Dias, A. A. Kassim, and V. Srinivasan. A neural network based corner detection method. In *Proc. IEEE Int'l Conference on Neural Networks (ICNN)*, volume 4, pages 2116–2120, Nov/Dec 1995.
- [34] A. Dimai. Rotation invariant texture description using general moment invariants and gabor filters. In *Proc. 11th Scandinavian Conference on Image Analysis (SCIA)*, volume 1, pages 391–398, June 1999.
- [35] A. Diplaros, T. Gevers, and I. Patras. Combining color and shape information for illumination-viewpoint invariant object recognition. *IEEE Trans. on Image Processing (TIP)*, 15(1):1–11, Jan 2006.
- [36] M. Donoser and H. Bischof. Efficient maximally stable extremal region (MSER) tracking. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 553–560, June 2006.
- [37] C. Enroth-Cugell and J. .G. Robson. The contrast sensitivity of retinal ganglion cells of the cat. *Journal of Physiology*, 187(3):517–552, Dec 1966.
- [38] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, Nov 1993.

- [39] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531, June 2005.
- [40] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. 10th IEEE Int'l Conference on Computer Vision (ICCV)*, volume 2, pages 1816–1823, Oct 2005.
- [41] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.
- [42] 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*, 4(2):171–187, May 1994.
- [43] J. J. Foo and R. Sinha. Pruning SIFT for scalable near-duplicate image matching. In *Proc. 18th Australasian Database Conference (ADC)*, volume 63, pages 63–71, Jan/Feb 2007.
- [44] P.-E. Forssén and D. G. Lowe. Shape descriptors for maximally stable extremal regions. In *Proc. 11th IEEE Int'l Conference on Computer Vision (ICCV)*, pages 1–8, Oct 2007.
- [45] K. Fredriksson and E. Ukkonen. Faster template matching without FFT. In *Proc. 8th IEEE Int'l Conference on Image Processing (ICIP)*, volume 1, pages 678–681, Oct 2001.
- [46] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 13(9):891–906, Sept 1991.
- [47] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. 8th European Conference on Computer Vision (ECCV)*, volume 3, pages 224–237, May 2004.

- [48] K. Fukunaga and W. L. G. Koontz. Application of the karhunen-loève expansion to feature selection and ordering. *IEEE Trans. on Computers*, 19(4):311–318, April 1970.
- [49] B. V. Funt, K. Barnard, and L. Martin. Is machine colour constancy good enough? In *Proc. 5th European Conference on Computer Vision (ECCV)*, volume 1, pages 445–459, June 1998.
- [50] T. Goedemé, T. Tuytelaars, and L. Van-Gool. Fast wide baseline matching for visual navigation. In *Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 24–29, June/July 2004.
- [51] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*, pages 263–284. The MIT Press, Cambridge, MA, 1990.
- [52] G. D. Hager and P. N. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–410, June 1996.
- [53] J. S. Hare and P. H. Lewis. Salient regions for query by image content. In *Proc. 3rd Int’l Conference on Image and Video Retrieval (CIVR)*, pages 317–325, July 2004.
- [54] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference (AVC)*, pages 147–151, Aug/Sept 1988.
- [55] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, UK, Sept 2000.
- [56] M. Herman and T. Kanade. Incremental reconstruction of 3D scenes from multiple, complex images. *Artificial Intelligence (AI)*, 30(3):289–342, Dec 1986.
- [57] P. V. C. Hough. Method and means for recognizing complex patterns. U.S. Patent 3069654, Dec 1962.

- [58] IANS. Machines still way behind humans in image recognition. *Communications of the ACM*, 10 Sept 2009. Available at: <http://cacm.acm.org/news/40791-machines-still-way-behind-humans-in-image-recognition/fulltext>, [Online; accessed 2 Jan 2010].
- [59] H. Inoue, T. Tachikawa, and M. Inaba. Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation. In *Proc. IEEE Int'l Conference on Robotics and Automation (ICRA)*, volume 2, pages 1621–1626, May 1992.
- [60] I. T. Jolliffe. Principal component analysis. Springer-Verlag, New York, 1986.
- [61] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proc. 8th European Conference on Computer Vision (ECCV)*, volume 1, pages 228–241, May 2004.
- [62] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception*. Praeger, 1979.
- [63] J. Kannala and S. S. Brandt. Quasi-dense wide baseline matching using match propagation. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.
- [64] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l Journal of Computer Vision (IJCV)*, 1(4):321–331, Jan 1988.
- [65] A. Katare, S. K. Mitra, and A. Banerjee. Content based image retrieval system for multi object images using combined features. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–599, June 2007.
- [66] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 506–513, June/July 2004.
- [67] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370, Aug 1984.

- [68] J. Košecká and W. Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding (CVIU)*, 100(3):274–293, Dec 2005.
- [69] T. Kyriacou, G. Bugmann, and S. Lauria. Vision-based urban navigation procedures for verbally instructed robots. *Robotics and Autonomous Systems*, 51(1):69–80, April 2005.
- [70] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(8):1265–1278, Aug 2005.
- [71] J. A. Lee and K.-C. Yow. Image recognition for mobile applications. In *Proc. 14th IEEE Int’l Conference on Image Processing (ICIP)*, volume 6, pages 177–180, Sept 2007.
- [72] J. A. Lee and K.-C. Yow. Combining invariant and corner-like features to optimize image matching. In *Proc. 3rd Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pages 692–701, Jan 2009.
- [73] J. A. Lee and K.-C. Yow. Optimizing image matches via a verification model. *Int’l Journal of Intelligent Systems (IJIS)*, 25(11):1103–1120, Nov 2010.
- [74] J. A. Lee and K.-C. Yow. Robust matching of building facades across widely separated views. *Image and Vision Computing (IVC)*, 2011. Under review.
- [75] J. A. Lee, K.-C. Yow, and Y.-S. Chia. Robust matching of building facades under large viewpoint changes. In *Proc. 12th IEEE Int’l Conference on Computer Vision (ICCV)*, pages 1258–1264, Sept/Oct 2009.
- [76] J. A. Lee, K.-C. Yow, and I. Kumazawa. A robust image retrieval system for mobile guide applications. *Int’l Journal of Intelligent Systems (IJIS)*, 2011. Under review.
- [77] J. A. Lee, K.-C. Yow, and A. Sluzek. Image-based information guide on mobile devices. In *Proc. 4th Int’l Symposium on Visual Computing (ISVC)*, pages 346–355, Dec 2008.

- [78] H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, and L. Amsaleg. Scalability of local image descriptors: a comparative study. In *Proc. 14th annual ACM Int'l conference on Multimedia (MULTIMEDIA)*, pages 589–598, Oct 2006.
- [79] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proc. IEEE/RSJ Int'l Workshop on Intelligent Robots and Systems (IROS)*, volume 3, pages 1442–1447, Nov 1991.
- [80] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.
- [81] M. K. Leung and Y.-H. Yang. Dynamic two-strip algorithm in curve fitting. *Pattern Recognition (PR)*, 23(1/2):69–79, Jan 1990.
- [82] J. P. Lewis. Fast template matching. In *Proc. Int'l Conference on Vision Interface (VI)*, pages 120–123, May 1995.
- [83] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.
- [84] T. Lindeberg. Feature detection with automatic scale selection. *Int'l Journal of Computer Vision (IJCV)*, 30(2):79–116, Nov 1998.
- [85] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image and Vision Computing (IVC)*, 15(6):415–434, June 1997.
- [86] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 246–253, June 2006.
- [87] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. 7th IEEE Int'l Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, Sept 1999.

- [88] D. G. Lowe. Local feature view clustering for 3D object recognition. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 682–688, Dec 2001.
- [89] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision (IJCV)*, 60(2):91–110, Nov 2004.
- [90] Z. Ma. *Artificial Intelligence for Maximizing Content Based Image Retrieval*, page 130. Idea Group Inc. (IGI) Global, Hershey PA, USA, Nov 2008.
- [91] P. C. Mahalanobis. On the generalised distance in statistics. In *Proc. National Institute of Sciences of India*, volume 2, pages 49–55, 1936.
- [92] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Proc. 13th British Machine Vision Conference (BMVC)*, pages 384–393, Sept 2002.
- [93] M. J. McMahon, O. S. Packer, and D. M. Dacey. The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-gabaergic pathway. *Journal of Neuroscience*, 24(15):3736–3745, April 2004.
- [94] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 26–36, June 2006.
- [95] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. 8th IEEE Int'l Conference on Computer Vision (ICCV)*, pages 525–531, July 2001.
- [96] K. Mikolajczyk and C. Schmid. An affine invariant interest points detector. In *Proc. 7th European Conference on Computer Vision (ECCV)*, pages 128–142, May/June 2002.
- [97] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int'l Journal of Computer Vision (IJCV)*, 60(1):63–86, Oct 2004.

- [98] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(10):1615–1630, Oct 2005.
- [99] K. Mikolajczyk, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van-Gool. A comparison of affine region detectors. *Int'l Journal of Computer Vision (IJCV)*, 65(1-2):43–72, Nov 2005.
- [100] F. Mindru, T. Tuytelaars, L. Van-Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding (CVIU)*, 94(3):3–27, June 2004.
- [101] T. M. Mitchell. *Machine Learning*. McGraw Hill, March 1997.
- [102] B. Mičušík, H. Wildenauer, and J. Košecká. Detection and matching of rectilinear structures. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, June 2008.
- [103] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. 5th Int'l Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, page 584, Aug 1977.
- [104] H. P. Moravec. Visual mapping by a robot rover. In *Proc. 6th Int'l Joint Conference on Artificial Intelligence (IJCAI)*, pages 598–600, Aug 1979.
- [105] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *Int'l Journal of Computer Vision (IJCV)*, 73(3):263–284, July 2007.
- [106] J.-M. Morel and G. Yu. On the consistency of the SIFT method. Technical report prepublication, inverse problems and imaging (IPI), CMLA, ENS Cachan, France, Sept 2008.
- [107] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Science*, 2(2):438–469, April 2009.
- [108] H. Murase and S. K. Nayar. Detection of 3D objects in cluttered scenes using hierarchical eigenspace. *Pattern Recognition Letters*, 18(4):375–384, April 1997.

- [109] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 11–18, June 2006.
- [110] D. R. Myatt, P. H. S. Torr, S. J. Nasuto, J. M. Bishop, and R. Craddock. NAPSAC: High noise, high dimensional robust estimation - it's in the bag. In *Proc. 13th British Machine Vision Conference (BMVC)*, volume 2, pages 458–467, Sept 2002.
- [111] V. S. Nalwa. Line-drawing interpretation: Straight lines and conic sections. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 10(4):514–529, July 1988.
- [112] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content, using color, texture, and shape. *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, 1908:173–187, Jan/Feb 1993.
- [113] D. Nistér. Preemptive RANSAC for live structure and motion estimation. In *Proc. 9th IEEE Int'l Conference on Computer Vision (ICCV)*, volume 1, pages 199–206, Oct 2003.
- [114] D. Nistér and H. Stewénus. Scalable recognition with a vocabulary tree. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168, June 2006.
- [115] D. Nistér and H. Stewénus. Linear time maximally stable extremal regions. In *Proc. 10th European Conference on Computer Vision (ECCV)*, pages 183–196, Oct 2008.
- [116] Š. Obdržálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proc. 13th British Machine Vision Conference (BMVC)*, pages 113–122, Sept 2002.
- [117] D. Parks and J-P. Gravel. Corner detectors. 2010. Available at: <http://www.cim.mcgill.ca/~dparks/CornerDetector/index.htm>, [Online; accessed 4 Jan 2010].

- [118] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. 6th IEEE Int'l Conference on Computer Vision (ICCV)*, pages 754–760, Jan 1998.
- [119] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.
- [120] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *Proc. 15th British Machine Vision Conference (BMVC)*, pages 819–828, Sept 2004.
- [121] P. I. Rockett. Performance assessment of feature detection algorithm: A methodology and case study on corner detectors. *IEEE Trans. on Image Processing (TIP)*, 12(12):1668–1676, Dec 2003.
- [122] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Proc. 10th IEEE Int'l Conference on Computer Vision (ICCV)*, volume 2, pages 1508–1511, Oct 2005.
- [123] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. 9th European Conference on Computer Vision (ECCV)*, volume 1, pages 430–443, May 2006.
- [124] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 32(1):105–119, Jan 2010.
- [125] E. Rosten, G. Reitmayr, and T. Drummond. Real-time video annotations for augmented reality. In *Proc. 1st Int'l Symposium on Visual Computing (ISVC)*, pages 294–302, Dec 2006.
- [126] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int'l Journal of Computer Vision (IJCV)*, 40(2):99–121, Nov 2000.
- [127] B. C. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman. Object recognition by scene alignment. In *Proc. 21st Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1241–1248, Dec 2007.

- [128] M. A. Ruzon and C. Tomasi. Edge, junction, and corner detection using color distributions. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1281–1295, Nov 2001.
- [129] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(5):824–840, May 2009.
- [130] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. 8th IEEE Int’l Conference on Computer Vision (ICCV)*, volume 2, pages 636–643, July 2001.
- [131] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or “How do i organize my holiday snaps?”. In *Proc. 7th European Conference on Computer Vision (ECCV)*, volume 1, pages 414–431, May/June 2002.
- [132] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, June 2008.
- [133] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int’l Journal of Computer Vision (IJCV)*, 37(2):151–172, June 2000.
- [134] S. Se, D. G. Lowe, and J. Little. Global localization using distinctive visual features. In *Proc. IEEE/RSJ Int’l Conference on Intelligent Robots and Systems (IROS)*, pages 226–231, Oct 2002.
- [135] H. Shao, T. Svoboda, and L. Van-Gool. Zubud-zurich buildings database for image based recognition. Technical Report 260, Computer Vision Lab, Swiss Federal Institute of Technology, Zurich, Switzerland, April 2003.
- [136] H. C. Shen and A. K. C. Wong. Generalized texture representation and metric. *Computer Vision, Graphics, and Image Processing (CVGIP)*, 23(2):187–206, Aug 1983.

- [137] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proc. 9th IEEE Int'l Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477, Oct 2003.
- [138] A. Sluzek. On moment-based local operators for detecting image patterns. *Image and Vision Computing (IVC)*, 23(3):287–298, March 2005.
- [139] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(11):1061–1075, Nov 1992.
- [140] I. Stamos and P. K. Allen. 3-D model construction using range and image data. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 531–536, June 2000.
- [141] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 259–266, June 2003.
- [142] G. P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 521–527, June 1999.
- [143] M. Stricker and M. Orengo. Similarity of color images. In *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 381–392, Feb 1995.
- [144] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing visual scenes using transformed objects and parts. *Int'l Journal on Computer Vision (IJCV)*, 77(1-3):291–330, May 2008.
- [145] C. Sun. A fast stereo matching method. In *Proc. Int'l Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 95–100, Dec 1997.
- [146] M. J. Swain and D. H. Ballard. Indexing via color histograms. In *Proc. 3rd IEEE Int'l Conference on Computer Vision (ICCV)*, pages 390–393, Dec 1990.

- [147] M. J. Swain and D. H. Ballard. Color indexing. *Int'l Journal of Computer Vision (IJCV)*, 7(1):11–32, Nov 1991.
- [148] D. Tell and S. Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *Proc. 6th European Conference on Computer Vision (ECCV)*, volume 1, pages 814–828, June/July 2000.
- [149] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *Proc. 7th European Conference on Computer Vision (ECCV)*, volume 1, pages 68–81, May/June 2002.
- [150] B. Tordoff and D. W. Murray. Guided sampling and consensus for motion estimation. In *Proc. 7th European Conference on Computer Vision (ECCV)*, volume 1, pages 82–96, May/June 2002.
- [151] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int'l Journal of Computer Vision (IJCV)*, 24(3):217–300, Oct 1997.
- [152] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding (CVIU)*, 78(1):138–156, April 2000.
- [153] P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding (CVIU)*, 71(3):312–333, Sept 1998.
- [154] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, June 1991.
- [155] T. Tuytelaars and L. Van-Gool. Content-based image retrieval based on local affinity invariant regions. In *Proc. 3rd Int'l Conference on Visual Information Systems (VISUAL)*, pages 493–500, June 1999.

- [156] T. Tuytelaars and L. Van-Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. 11th British Machine Vision Conference (BMVC)*, pages 412–425, Sept 2000.
- [157] F. Ullah and S. Kaneko. Using orientation codes for rotation-invariant template matching. *Pattern Recognition (PR)*, 37(2):201–209, Feb 2004.
- [158] M. Urschler, J. Bauer, H. Ditt, and H. Bischof. SIFT and shape context for feature-based nonlinear registration of thoracic CT images. In *Proc. 2nd Int’l Workshop on Computer Vision Approaches to Medical Image Analysis (CVAMIA), held in conjunction with 9th European Conference on Computer Vision (ECCV) 2006*, pages 73–84, May 2006.
- [159] J. van de Weijer, T. Gevers, and A. D. Bagdanov. Boosting color saliency in image feature detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(1):150–156, Jan 2006.
- [160] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *Proc. 9th European Conference on Computer Vision (ECCV)*, volume 2, pages 334–348, May 2006.
- [161] Visual Geometry Group, Oxford University. Available at: <http://www.robots.ox.ac.uk/~vgg/data/>, 2010.
- [162] C. Y. Wang. *Edge detection using template matching*. PhD dissertation, School of Engineering, Duke University, Jan 1985.
- [163] J. Wang, H. Zha, and R. Cipolla. Combining interest points and edges for content-based image retrieval. In *Proc. 12th IEEE Int’l Conference on Image Processing (ICIP)*, volume 3, pages 1256–1259, Sept 2005.
- [164] L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *Proc. 12th IEEE Int’l Conference on Computer Vision (ICCV)*, pages 1258–1264, Sept/Oct 2009.

- [165] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms. *Computer Vision, Graphics, and Image Processing (CVGIP)*, 32(3):328–336, Dec 1985.
- [166] M. Wertheimer. Laws of organization in perceptual forms. In W. D. Ellis, editor, *A source book of Gestalt psychology*, pages 71–88. London: Routledge and Kegan Paul, 1938.
- [167] Wikipedia. Cross-correlation — wikipedia, the free encyclopedia, 2010. Available at: <http://en.wikipedia.org/w/index.php?title=Cross-correlation&oldid=358163101>, [Online; accessed 5 May 2010].
- [168] Wikipedia. Minkowski distance — wikipedia, the free encyclopedia, 2010. Available at: http://en.wikipedia.org/w/index.php?title=Minkowski_distance&oldid=357465752, [Online; accessed 2 May 2010].
- [169] Wikipedia. Pearson’s chi-square test — wikipedia, the free encyclopedia, 2010. Available at: http://en.wikipedia.org/w/index.php?title=Pearson%27s_chi-square_test&oldid=356063955, [Online; accessed 3 May 2010].
- [170] Wikipedia. Scale-invariant feature transform — wikipedia, the free encyclopedia, 2010. Available at: http://en.wikipedia.org/w/index.php?title=Scale-invariant_feature_transform&oldid=341785838, [Online; accessed 5 Feb 2010].
- [171] A. P. Witkin. Scale-space filtering. In *Proc. 8th Int’l Joint Conference on Artificial Intelligence (IJCAI)*, pages 1019–1022, Aug 1983.
- [172] J. Xiao and M. Shah. Two-frame wide baseline matching. In *Proc. 9th IEEE Int’l Conference on Computer Vision (ICCV)*, volume 2, pages 603–609, Oct 2003.
- [173] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(11):1973–1989, Nov 2007.

- [174] R. A. Young. The gaussian derivative model for spatial vision: I. retinal mechanisms. *Spatial Vision*, 2(4):273–293, 1987.
- [175] K.-C. Yow and J. A. Lee. Mobile tourguide system. In *Proc. Singaporean-French IPAL Symposium (SinFra)*, pages 116–124, Feb 2009.
- [176] G. Yu and J.-M. Morel. A fully affine invariant image comparison method. In *Proc. IEEE Int’l Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1597–1600, April 2009.
- [177] W. Zhang and J. Kořecká. Image based localization in urban environments. In *Proc. 3rd Int’l Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 33–40, June 2006.
- [178] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *Int’l Journal of Computer Vision (IJCV)*, 27(2):161–195, March 1998.
- [179] D. Zwillinger. *CRC Standard Mathematical Tables and Formulae*, page 223. CRC Press, Inc., Boca Raton, FL, 30th edition, 1996.

List of Publications

Journal

1. Jimmy Addison Lee and Kin-Choong Yow, Optimizing Image Matches via a Verification Model, *Int'l Journal of Intelligent Systems (IJIS)*, 25(11):1103-1120, Nov 2010.
2. Jimmy Addison Lee and Kin-Choong Yow, Robust Matching of Building Facades across Widely Separated Views, *Image and Vision Computing (IVC)*, pages unknown, 2011. (Under review).
3. Jimmy Addison Lee, Kin-Choong Yow, and Itsuo Kumazawa, A Robust Image Retrieval System for Mobile Guide Applications, *Int'l Journal of Intelligent Systems (IJIS)*, pages unknown, 2011. (Under review).

Conference

1. Jimmy Addison Lee and Kin-Choong Yow, Locating Corresponding Regions in Urban Environments, *In Proc. 4th Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pages 33-38, Nov 2010.
2. Jimmy Addison Lee, Kin-Choong Yow, and Alex Yong-Sang Chia, Robust Matching of Building Facades under Large Viewpoint Changes, *In Proc. 12th IEEE Int'l Conference on Computer Vision (ICCV)*, pages 1258-1264, Sept 2009.
3. Jimmy Addison Lee and Kin-Choong Yow, Combining invariant and corner-like features to optimize image matching, *In Proc. 3rd Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pages 692-701, Jan 2009.

4. Kin-Choong Yow and Jimmy Addison Lee, Mobile Tourguide System, *In Proc. Singaporean-French IPAL (Image Perception, Access and Language) Symposium (Sin-Fra)*, pages 116-124, Feb 2009.
5. Jimmy Addison Lee, Kin-Choong Yow, and Andrzej Sluzek, Image-Based Information Guide on Mobile Devices, *In Proc. 4th Int'l Symposium on Visual Computing (ISVC)*, pages 346-355, Dec 2008.
6. Jimmy Addison Lee and Kin-Choong Yow, Image Recognition for Mobile Applications, *In Proc. 14th IEEE Int'l Conference on Image Processing (ICIP)*, vol. 6, pages 177-180, Sept 2007.