Improving the Effectiveness of Local Feature Detection

Shoaib Ehsan

A thesis submitted for the degree of

Doctor of Philosophy

at the

School of Computer Science and Electronic Engineering

University of Essex

May 2012

Abstract

The last few years have seen the emergence of sophisticated computer vision systems that target complex real-world problems. Although several factors have contributed to this success, the ability to solve the image correspondence problem by utilizing local image features in the presence of various image transformations has made a major contribution.

The trend of solving the image correspondence problem by a three-stage system that comprises detection, description, and matching is prevalent in most vision systems today. This thesis concentrates on improving the local feature detection part of this three-stage pipeline, generally targeting the image correspondence problem.

The thesis presents offline and online performance metrics that reflect real-world performance for local feature detectors and shows how they can be utilized for building more effective vision systems, confirming in a statistically meaningful way that these metrics work.

It then shows how knowledge of individual feature detectors' functions allows them to be combined and made into an integral part of a robust vision system. Several improvements to feature detectors' performance in terms of matching accuracy and speed of execution are presented.

Finally, the thesis demonstrates how resource-efficient architectures can be designed for local feature detection methods, especially for embedded vision applications.

To Abbu, Ammi, Maria & Guria

Acknowledgements

First of all, I would like to thank ALLAH Almighty for giving me the strength and determination to complete this work. The last four years have been miraculous for me due to the mercy of ALLAH Almighty; I don't have the words to express my gratitude.

Without any doubt, this thesis will be incomplete without highlighting the great supervision and support of Prof. Klaus McDonald-Maier and Dr. Adrian Clark. To be honest, this thesis is a product of our team work. Not only I learnt the art of research and academic writing from them, Klaus and Adrian are hugely responsible for transforming my outlook as a researcher and always motivated me to achieve the highest research standards. Thanks to them for being so patient with me during those countless enlightening discussions. I could not have accomplished so much without their unlimited and untiring support. I must also appreciate the contribution of Adrian in capturing all those large image databases.

Many thanks to Dr. David Hunter for being on my supervisory board and for all the encouragement. The guidance I received from Dr. Tom Foulsham towards the end of my research really deserves high appreciation. I am also thankful to Prof. Roy Davies for being very compassionate and supportive to me. I will forever remember Prof. Atta Badii as a person who was always there for me whenever I needed help. I am really grateful to Dr. Andrew Hopkins for being a great inspiration for me throughout my PhD years. I also wish to acknowledge the great support of Sheineez (FunTech) during the early stages of this work.

I really appreciate the never ending support of my parents. Whatever I am today is due to their determined efforts. I cannot pay them back for all their love, care and prayers. I feel obliged to my grandparents. I also wish to thank my parents-in-law. My wife Maria deserves a special mention as she has backed me through all tough times. I definitely owe her a lot. Aila, my

beloved daughter, has been a source of joy and comfort all along. The smiles of my dearly loved son Abdullah have always helped me to relax. I would also like to acknowledge the support of my sisters, Nadia, Rabia and Ayesha. Thanks to Naveed *bhai*, Rizwan *bhai*, Usman *bhai*, Jahangir *bhai* and their families. I am also grateful to Ikram uncle, Najma aunty and Rasheeda aunty.

I do not have words to express gratitude for Farooq *bhai*, Aqsa *bhabhi* and Naima. They helped me and my family in every possible way and made our stay here really wonderful. I will always remember the good times that we spent together.

Big thanks to all my colleagues, especially Nadia Kanwal, Erkan Bostanci, Yasir Qadri, Philip Cheung and Yevgeniya Kovalchuk. I am really obliged to Bayar Menzat and Dragos Stanciu.

Thanks is also due for Marisa Bostock, Ramona Bacon, Jayne Bates, Simon Moore, Alfonso Torrejon, Maria Ntovrou, Alex Schillings, David Himsworth, Cristian Juganaru, Atanas Kuzmanov and David Snow.

My stay in the UK would not have been as pleasurable as it was, had it not been for family-friends like Waqar Nabi, Tahseen Waqar, Dr. Sajid Baloch, Fowad Murtaza, Saima Fowad, Dr. Zahid Waheed, Ammara Zahid, Yasir Qadri, Nadia Qadri, Nadia Kanwal, Tahseen Muzaffar, Erkan Bostanci, Betul Bostanci, Mamoona Asghar, Zain-ul-Abdin Khuhro, Farhat Memon, Tahir Qadri and Ayesha Tahir.

My long-time colleagues and friends also deserve credit: Junaid Afzal, Umar Hamid, Naveed-ur-Rehman, Arslan Khan, Shakaiba Majeed, Hamid Jabbar, Saima Siddiqi, Imran Siddiqi, Omer Zaman, Ahsen Javed, Sammar Javed and Sahar Javaid. Thanks for all your help and support over the years.

Special thanks to the University of Essex and EPSRC for providing the financial support for this work. I am grateful to the University of Essex, BMVA and ICIAR committee for awarding conference travel grants.

Contents

List of Figures

List of Tables

Abbreviations

SIFT	Scale Invariant Feature Transform
SURF	Speeded-Up Robust Features
EBR	Edge-Based Regions
IBR	Intensity-Based Regions
MSER	Maximally Stable Extremal Regions
SFOP	Scale Invariant Feature Operator
HAR-LAP	Harris-Laplace
HES-LAP	Hessian-Laplace
HAR-AFF	Harris-Affine
HES-AFF	Hessian-Affine
LoG	Laplacian of Gaussian
DoG	Difference-of-Gaussians
FAST	Features from Accelerated Segment Test
HVS	Human Visual System
ROC	Receiver Operating Characteristic
GPU	Graphics Processing Unit
HD	High Definition

1 Introduction

The important thing is not to stop questioning.

ALBERT EINSTEIN

From automated panorama creation to the more exciting applications like image-based place recognition and modeling the world from internet photo collections, local invariant feature detection finds itself at the heart of many sophisticated vision systems today. Despite significant advances in the last decade or so, the quest for more robust feature detection methods continues so as to make vision systems more effective and reliable. This chapter takes a look at some of the current challenges in this domain to elucidate the motivation behind this work. The major contributions made by this thesis in an attempt to bridge these research gaps are also highlighted. A snapshot of each chapter is presented to illustrate the structure of the thesis. Finally, publications that were made during the course of this research are listed to end the chapter.

1.1 The Renaissance of Local Feature Detection

> *Feature point detectors and descriptors are the most important recent advance in computer vision and graphics.*
>
> [1]WILLIAM T. FREEMAN, MIT

The brevity of the above statement is incredibly striking; William T. Freeman seems to have hit the nail on the head. Indeed, the ability to solve the image correspondence problem with reasonable accuracy by detecting, describing and matching local features under various geometric and photometric transformations, such as viewpoint changes and illumination variations, has revolutionized the entire field of computer vision over the past decade by enabling novel applications and sophisticated vision systems. While a large body of literature already exists regarding correspondence from local image features [1], this thesis represents an attempt to accentuate its importance by pushing the boundaries further.

Local features have been around since the mid-1950s [2]. The initial era (1954-1998) saw the advent of some promising feature detectors, such as Moravec [3], Harris and Stephens (widely known as the Harris detector today) [4], Beaudet operators [5], Kitchen and Rosenfeld [6], Dreschler and Nagel [7], Forstner operators [8, 9] and SUSAN [10]. However, it was the emergence of the Scale Invariant Feature Transform (SIFT) [11, 12], a fast local invariant feature detector coupled with a highly distinctive descriptor for solving the image correspondence problem, that invigorated the interest of the vision community in local features. So massive has been the impact of SIFT that the next ten years saw a large number of techniques based on local invariant features being proposed [1, 13-25] utilizing the same basic framework. With more than 12300 citations on Google Scholar to date, SIFT has indubitably served as a stimulant for revival of local feature detection over the past decade.

---

[1] 'Where Machine Vision Needs Help From Machine Learning', Keynote speech at the 24th Annual Conference on Learning Theory, Budapest, Hungary, July 9-11, 2011.

A reflection of the achievement of SIFT and its descendants is the huge number of applications and sophisticated vision systems that have been developed utilizing the platform provided by these techniques. Automatic panorama creation [26, 27], wide baseline matching for stereo pairs [14, 28], object recognition [11], image retrieval from large databases [29], object retrieval in video [30], image-based place recognition [31], object categorization [32-35], symmetry detection [36], robot localization [37], shot location [38], texture recognition [39, 40], hand gesture recognition [41] and modeling the world from internet photo collections [42] are some of the application areas in which local invariant features have been utilized successfully.

Robust algorithms for all the three stages, namely detection, description and matching, are fundamental to the success of any system aiming to solve the image correspondence problem utilizing local features. Although these steps are not completely independent and essentially constitute a pipelined system where the output of one stage serves as the input for the other (see Figure 1-1), it is generally assumed in the literature that these stages are independent. This assumption helps to avoid a complicated system where it becomes hard to separate the effect of the different stages. The same approach is followed in this thesis. While generally targeting the image correspondence problem, this thesis mainly concentrates on the detection step for improving the effectiveness and reliability of the three-stage system shown in Figure 1-1. For feature description and matching, the author employs the SIFT descriptor [11, 12] and the Nearest Neighbor matching scheme [12] respectively throughout this thesis unless stated otherwise.

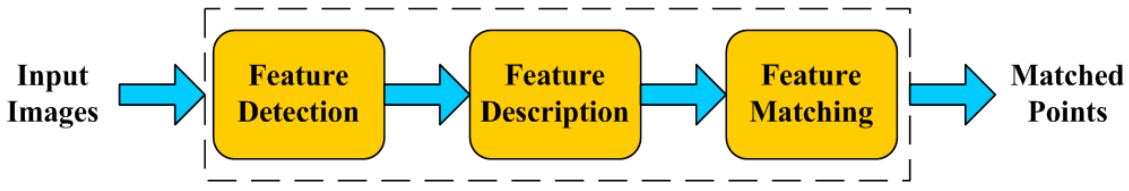

Figure 1-1: A simplified block diagram of an image matching system based on local features

1.2 Challenges

While it is true that great strides have been made in the field of local invariant feature detection during the last few years, there are still many avenues that need to be pursued and remain challenging for vision researchers. Here, those open research issues are mentioned which provide the impetus for the work presented in this thesis:

1) *Accuracy*. An important driving factor in this research domain is the improvement of the accuracy of the image matching system. True matched points increase the accuracy of the system whereas false matched points have a negative effect on it. Although all three stages shown in Figure 1-1 are important in this regard, the role of feature detection is critical for being the initial step. How to select adequate suitable features that can provide sufficient true matches, ideally with no false matches, to maximize the accuracy of the system still remains an open question [1].

2) *Reliable performance indicators*. Vision-based systems usually operate in complex and unknown environments across a range of different applications. It is a challenging task to predict the performance of such a system without prior knowledge of imaging conditions. The metrics which are currently available to gauge the performance of feature detectors do not always reflect actual performance [1]. Offline indicators which would provide accurate and reliable insight into the behavior of a feature detector and the system as a whole, before deployment in the actual environment, would be of great value.

3) *Complementarity of feature detectors*. To tackle the uncertainty of image content, running multiple feature detectors simultaneously for solving complex vision problems is an emerging trend [1, 30, 43, 44]. However, it has detrimental effects on overall computation time due

to the increasing amount of data that need to be processed, and usually provides an overcomplete representation of an image rather than a compact one. Moreover, combining multiple detectors may have adverse effects on their combined performance, in some cases even making it lower than what can be achieved by a single detector [39, 45]. It is therefore desirable to explore and define different complementarity measures that identify detector combinations capable of achieving better performance than the individual detectors. Another requirement is to define a generic framework that can decide automatically when to employ multiple feature detectors in parallel and when to operate in single detector mode depending upon the image content. Such a framework would be really useful for time-critical applications.

4) *Online performance analysis*. Existing performance measures for local feature detectors allow only offline assessment due to their requirement for ground truth data and high computation cost [19, 46]. As mentioned earlier, the environments in which the vision systems are deployed are usually unknown and complex. A feature detector that can gauge its own performance and take appropriate actions online to maximize its effectiveness would be valuable. It would allow the feature detector to adapt to the nature of the imagery it is processing. Performance metrics that can be computed quickly are therefore required to achieve the goal of online performance analysis.

5) *Evaluation frameworks and datasets*. Indubitably, the availability of a wide variety of local invariant feature detectors today has rendered the task of evaluating them an important issue in vision research [47, 48]. Unbiased dataset collection and careful design of algorithm evaluation protocols are critical for finding the strengths and weaknesses of competing techniques accurately [49]. From a vision systems design perspective, evaluation frameworks that allow

identification of statistically significant performance differences between feature detectors would be valuable. Similarly, datasets that would permit cross-dataset generalization of results would be really useful.

6) *Real-time systems*. Generally, image processing and computer vision algorithms are computation- and data-intensive in nature. The same applies to local invariant feature detectors. Many state-of-the-art feature detection techniques require prohibitively long processing times, ranging from a few seconds to a few minutes, making them unsuitable for real-time applications on commodity hardware. As local feature detection has reached some level of maturity now, the next challenge is to explore methods to reduce this computation and design optimized algorithms that allow feature detectors to achieve real-time performance.

7) *Embedded Vision Systems*. We are in the midst of an imaging revolution. Inexpensive digital cameras have made the spectrum of embedded vision applications broader and broader. Cell phones and robots are the most common platforms for such vision-based systems. As embedded systems generally have strict constraints on power consumption, memory size, chip area and weight, it is a challenging task to run computation-intensive feature detection algorithms on such systems. Progress needs to be made both at hardware and software levels to allow state-of-the-art feature detectors to be employed in embedded vision applications.

8) *Parallel algorithms*. Due to the advent of multi-core architectures, parallel algorithms are becoming more important than ever before. It is no surprise that most existing feature detection algorithms are serial in nature. Deigning parallel algorithms for feature detection would help in achieving the goal of real-time processing. Moreover, such algorithms would be useful for hardware acceleration.

1.3 Thesis Contributions

The contributions made during the course of this research are outlined below.

- Improved repeatability measures are proposed for performance characterization of local feature detectors which correlate much better with the true performance of feature detectors than the original repeatability metric [15, 46]. Evaluation results based on the proposed measures are presented for eleven state-of-the-art local feature detectors utilizing the widely-used Oxford datasets [50].

- In an effort to make the improved repeatability measures useful from a systems design viewpoint, a novel generic framework is presented which estimates the upper and lower bounds of detector performance and finds statistically-significant performance differences between detectors as a function of image transformation amount by introducing a new variant of McNemar's test [51, 52]. The novel concept of segmenting the performance of feature detectors into *operating* and *guarantee* regions is also presented.

- Three new image databases are proposed for JPEG compression (7546 images with 539 different scenes), blur (5390 images with 539 different scenes) and uniform illumination changes (7546 images with 539 different scenes). Employing these databases, the utility of the above-mentioned generic framework is demonstrated by presenting results for several state-of-the-art detectors under JPEG compression, blur and uniform illumination changes. These results provide new insights into the behavior of feature detectors under these image transformations.

- To improve the performance of feature detectors in the presence of uniform light variations, the inclusion of a simple pre-processing step as part of any feature detection scheme is proposed. Results are

presented for several state-of-the-art detectors that confirm the utility of the proposed method.

- For carrying out an online performance analysis of a local feature detector, a method is presented which is based on the spatial distribution of its detected features. Utilizing this technique, results are presented for several state-of-the-art local feature detectors for the widely-used Oxford datasets [50]. Four new image datasets (Campus-1, Campus-2, Snow and Indoor) of more than 100 images each are also presented which are employed to find statistically-significant performance differences between different detectors.

- A method is proposed for performing an online complementarity analysis of local feature detectors based on the spatial distribution of their detected features. Utilizing Campus-1, Campus-2, Snow and Indoor datasets, pairs and triplets of detectors are identified that provide good performance in terms of spatial distribution of feature points.

- For combining feature detectors intelligently in vision applications that require reasonable distribution of feature points, a novel prediction-based framework is proposed. Based on the image content, it decides automatically when to employ multiple feature detectors and when to operate in single detector mode.

- An algorithm is presented for the selection of scale-space octaves for the Speeded-Up Robust Features (SURF) technique. This algorithm outperforms non-uniformly sampled SURF variants (with 2, 3 and 4 octaves) in terms of both matching performance and computation.

- To compute integral image in hardware with low computational resources, a parallel algorithm is proposed which processes two rows of an input image simultaneously.

- With the objective of speeding up the computation of integral image more in hardware, another parallel algorithm is presented that processes four rows of an input image simultaneously and consumes low computational resources.

- An efficient design strategy is proposed that reduces the internal memory requirements of a parallel integral image computation engine.

- To reduce the memory requirements significantly for the storage of integral image, an algorithm is presented that works even if the size of the filter to be computed (using an integral image) is almost equal to the size of the input image.

- Another method is proposed that allows substantial reduction in the memory requirements for storing an integral image in situations where the size of the filter to be computed (using the integral image) is much smaller than the size of the input image.

1.4 Thesis Structure

A brief outline of the material presented in this thesis is as follows.

Chapter 2 introduces local invariant feature detection and presents a literature survey encompassing the major advancements in this domain. The chapter covers state-of-the-art detectors, such as SIFT and Harris-Laplace, in more detail as they form the primary stage of many sophisticated vision systems today and are frequently employed by researchers in the vision community. Finally, the chapter provides an overview of the progress made in efficient hardware implementation of popular local invariant feature detectors.

Chapter 3 begins with a brief review of the methods used for performance characterization of local feature detectors. It then identifies the limitations of repeatability, the most frequently-employed theoretical metric

in this regard. In this chapter, improved repeatability measures are proposed which correlate much better with the true performance of feature detectors. An evaluation of several state-of-the-art feature detectors based on the presented measures utilizing widely-used image datasets is then carried out to finish this chapter.

Chapter 4 illustrates how the improved repeatability measures from the previous chapter can be utilized in the design of more reliable and effective vision systems. In this regard, it presents a generic framework that allows assessment of the upper and lower bounds of detector performance and finds statistically-significant performance differences between detectors as a function of image transformation amount by introducing a new variant of McNemar's test [51, 52]. In order to demonstrate the utility of the proposed framework, results for several state-of-the-art detectors are presented in this chapter using newly acquired, large image databases for JPEG compression, uniform light changes and blurring. Finally, this chapter proposes to include a pre-processing step as part of the detection scheme which improves the performance of state-of-the-art detectors significantly in the presence of uniform light variations.

Chapter 5 concentrates on the issue of online performance analysis of local feature detectors. It emphasizes that spatial distribution of local image features can be used as a good performance indicator and presents a metric that can be calculated rapidly, concurs with human visual assessments and is complementary to existing offline measures such as repeatability. Utilizing the proposed measure, the chapter presents qualitative results for several state-of-the-art detectors on widely-used datasets. A newly-acquired, larger image database is then used to identify statistically-significant performance differences between competing feature detectors. The chapter also proposes a measure of complementarity for combinations of detectors, correctly reflecting the underlying principles of individual detectors. Based on this metric, various detector pairs and

triplets are investigated quantitatively and the results provide a useful guideline for combining detectors in applications that require reasonable spatial distribution of image features. The chapter also details the timing results for the proposed metrics with different feature detectors to demonstrate their utility for online applications. Finally, a prediction-based framework for combining local feature detectors in vision applications is presented.

Chapter 6 covers the problem of scale-space octave selection for Speeded-Up Robust Features (SURF) algorithm. It discusses the effect of octave reduction on the matching performance of SURF. The chapter shows that discarding the higher scale-space octaves for reducing computation is not always the most sensible approach and presents an algorithm for choosing which octaves to discard based on properties of the imagery. Results presented in this chapter demonstrate the effectiveness of this "best octaves" algorithm both in terms of matching performance and computation.

Chapter 7 focuses on the integral image, an intermediate image representation that allows multi-scale feature detection techniques like SURF to achieve significant speed-ups. The chapter deals with two important problems regarding integral image: computation and storage. Two parallel algorithms are presented for speeding up the computation of integral images in hardware with low computational resources. By extending these algorithms, an efficient design strategy for reducing the internal memory requirements of the integral image computation unit is proposed. Regarding storage, two methods are presented that allow significant reduction in the memory requirements of integral image.

Chapter 8 provides a summary of the work presented in this thesis and draws important conclusions. Some promising directions for future research based on the work presented in this thesis are also identified. The chapter is finished with the closing remarks.

1.5 List of Publication

Following publications were made during the course of this work:

I. Ehsan, S., Kanwal, N., Clark, A. F. and McDonald-Maier, K. D., "*An Algorithm for the Contextual Adaption of SURF Octave Selection with Good Matching Performance: Best Octaves*," IEEE Transactions on Image Processing, vol. 21, No. 1, pp. 297-304, January 2012.

II. [2]Ehsan, S., Kanwal, N., Clark, A. F. and McDonald-Maier, K. D., "*Improved Repeatability Measures for Evaluating Performance of Feature Detectors,*" Electronics Letters, vol. 46, issue 14, pp. 998-1000, July 2010.

III. Ehsan, S., Clark, A. F., Cheung, W. M., Bais, A. M., Menzat, B. I., Kanwal, N. and McDonald-Maier, K. D., "*Memory-Efficient Design Strategy for a Parallel Embedded Integral Image Computation Engine*," Proceedings of the 15th Irish Machine Vision and Image Processing Conference (IMVIP), Dublin, Ireland, September 2011.

IV. [3]Ehsan, S., Kanwal, N., Clark, A. F. and McDonald-Maier, K. D., "*Measuring the Coverage of Interest Point Detectors*," Proceedings of the 8th International Conference on Image Analysis and Recognition (ICIAR), Part I, LNCS 6753, pp. 253-261, British Columbia, Canada, June 2011.

V. Ehsan, S., Kanwal, N., Bostanci, E., Clark, A. F. and McDonald-Maier, K. D., "*Analysis of Interest Point Distribution in SURF Octaves*," Proceedings of the 3rd International Conference on Machine Vision (ICMV), Hong Kong, December 2010.

[2] This paper was also selected as a featured article out of 46 papers published in that particular issue.
[3] This paper was the winner of ICIAR Student Travel Grant competition; it also received travel grant from the British Machine Vision Association.

VI. Ehsan, S. and McDonald-Maier, K. D., "*Exploring Integral Image Word Length Reduction Techniques for SURF Detector*," Proceedings of the 2nd International Conference on Computer and Electrical Engineering (ICCEE), vol. 1, pp. 635-639, Dubai, UAE, December 2009.

VII. Ehsan, S., Clark, A. F. and McDonald-Maier, K. D., "*Hardware Based Scale- and Rotation-Invariant Feature Extraction: A Retrospective Analysis and Future Directions*," Proceedings of the 2nd International Conference on Computer and Electrical Engineering (ICCEE), vol. 1, pp. 620-624, Dubai, UAE, December 2009.

VIII. [4]Ehsan, S., Clark, A. F. and McDonald-Maier, K. D., "*Novel Hardware Algorithms for Row-Parallel Integral Image Calculation*," Proceedings of Digital Image Computing: Techniques and Applications (DICTA), pp. 61-65, Melbourne, Australia, December 2009.

IX. Ehsan, S. and McDonald-Maier, K. D., "*On-Board Vision Processing for Small UAVs: Time to Rethink Strategy*," Proceedings of NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 75-81, California, USA, July 2009.

X. Kanwal, N., Ehsan, S., Bostanci, E. and Clark, A. F., "*Evaluating the Angular Sensitivity of Corner Detectors*," Proceedings of the 4th IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS), Ottawa, Canada, September 2011.

XI. Kanwal, N., Ehsan, S., Bostanci, E. and Clark, A. F., "*A Statistical Approach for Comparing the Performances of Corner Detectors*," Proceedings of the IEEE Pacific Rim Conference on Communications,

[4] This paper was the winner of the University of Essex Travel Grant competition.

Computers and Signal Processing, pp. 321-326, British Columbia, Canada, August 2011.

XII. Kanwal, N., Ehsan, S. and Clark, A. F., "*Are Performance Differences of Interest Operators Statistically Significant?*," Proceedings of the 14th International Conference on Computer Analysis of Images and Patterns (CAIP), Part II, LNCS 6855, pp. 429-436, Seville, Spain, August 2011.

XIII. Bostanci, E., Kanwal, N., Ehsan, S. and Clark, A. F., "*Tracking Methods for Augmented Reality*," Proceedings of the 3rd International Conference on Machine Vision (ICMV), Hong Kong, December 2010.

2 Local Invariant Feature Detection: A Review

If I have seen further it is only by standing on the shoulders of giants.

SIR ISAAC NEWTON

Although local features have been around for a long time, it was only towards the end of 1990s that researchers became interested in local invariant feature detection. This chapter provides a bird's eye view of the earlier work and the major advances in this domain during the last decade. Feature detection techniques like SIFT and SURF, which had a significant impact on the research field and form the primary stage of many computer vision systems today, are discussed in more detail. Since computer vision continues to strive towards real-time systems, an overview of the progress made in efficient hardware implementation of feature detectors is also provided.

2.1 Introduction

As mentioned in Chapter 1, the arrival of SIFT [11, 12] at the turn of the 21st century signalled the dawn of a new era in the short history of computer vision and has revolutionized the entire field since then. The interest of vision researchers in local invariant feature detection has continued to grow and numerous feature detectors have been proposed to solve the image correspondence problem under various geometric and photometric transformations. SURF [13, 53], SFOP [16] and MSER [14] are some of the obvious examples which had a particularly significant impact on the research field. Although each chapter in this thesis discusses the related work regarding the specific problem that is investigated in that particular chapter, it is worth having a look at the important developments in the domain of local invariant feature detection from a historical perspective.

A general introduction to the field of local invariant feature detection is provided in Section 2.2. This is followed by an overview of the literature on local feature detection, encompassing the early work on local features and the more recent advances in Section 2.3. The chapter discusses the state-of-the-art feature detection techniques, such as the ones mentioned above, in more detail in Section 2.4 as they form the primary stage of many sophisticated vision systems today. Section 2.5 provides an overview of the work done so far regarding hardware acceleration of local feature detectors. Finally, a summary of the chapter is presented in Section 2.6.

2.2 Local Invariant Features

A local feature is generally defined as an image pattern which is different from its immediate neighborhood and related with a variation of an image property or several properties (such as intensity, color and texture) simultaneously [1]. It is usually considered robust to occlusion and clutter. Corners, blobs and edges are some of the examples of local features. By utilizing the information in a region centered on a detected local feature, a descriptor is computed which is then used for image matching. SIFT [12],

SURF [13], PCA-SIFT [54] and GLOH [55] are some of the popular descriptors that are utilized in combination with most local feature detection techniques.

Since a vision system may encounter images having different types of geometric and photometric transformations, such as viewpoint and illumination changes, it is important that the employed local feature detector is not affected by such variations in imaging conditions. This essentially means that the local feature detector must be invariant to various image transformations to work reliably. For example, scale invariance and rotation invariance ideally imply that the same image features can still be extracted by the detector if the input image is scaled up or down by any scale factor and rotated by any angle.

A local invariant feature detector providing a set of well-localized and individually identifiable anchor points is especially suitable for matching, tracking, camera calibration, 3D reconstruction, pose estimation and image alignment applications [1]. Moreover, the detected features can also be used as a robust image representation for recognition tasks, such as objects and scenes, texture analysis, image retrieval, scene classification and video mining without the need for image segmentation [1].

Some desirable properties of local invariant features stated in [1] are: a high percentage of the detected features should be repeatable in two images of the same scene taken under different imaging conditions (useful for all types of applications); the features should have a lot of variation in the underlying intensity patterns as it helps with descriptor-based feature matching; depending upon the requirement of a specific application, a sufficient number of features should be detected (especially for class-level object or scene recognition methods that require large number of features); the localization accuracy of the detected features in image location and scale should be good (particularly useful for wide baseline matching, registration and structure from motion applications); and the features should be detected in a time-efficient manner (valuable for real-time applications).

2.3 A Very Brief History of Local Feature Detection

This section provides an overview of the literature on local feature detection, encompassing the early work in this domain and the more recent advances, from a historical perspective by categorizing feature detection methods into specific classes; see [1] for an in-depth exposition. The pioneering work on local features appeared back in 1954 when it was observed by [2] that information on shape is concentrated at dominant points having high curvature. Since then, the field has continued to grow rapidly and a large number of local feature detectors have emerged.

Many vision researchers have examined the curvature of contours to detect corners. During the 1970s and the 1980s, the methods based on contour curvature were quite popular and were mainly utilized for line drawings, piecewise constant regions and cad-cam images. Some methods that are representative of this class of local feature detectors are [6, 56-84].

Another category of local feature detectors are the intensity-based methods which directly analyze the image intensities, such as the techniques based on the first- and second-order gray-value derivatives. These methods are generally suitable for a wide variety of images. The techniques proposed by [3-9, 15, 17, 18, 21, 23, 85-98] are some of the methods that had a particularly significant impact within this specific category of local feature detectors.

Some vision researchers working on local features have also proposed methods in the context of artificial intelligence and visual recognition to model the processes of human brain. Such techniques are usually termed 'the biologically plausible methods'. Some representative techniques for this particular class are [99-108].

For extracting local features, the color information has also been exploited by different methods. The biologically plausible techniques usually utilize color information for creating saliency maps. The techniques presented by [105, 108-113] signify some of the important developments in

this specific category. Model-based methods form another group of local feature detection techniques. Some methods that are representative of this class are [114-121].

During the last decade or so, local feature detection methods that are invariant to various image transformations, such as viewpoint and scale changes, have become quite popular. This particular category of detectors is useful for a wide range of vision applications. The techniques described in [12-16, 18, 122] are some of the important methods presented so far.

Vision researchers have also employed segmentation-based methods for detecting local image features. The techniques presented by [14, 123-129] represent some of the key developments in this specific category. Finally, machine learning based techniques have also forced their way into local feature detection domain. Some examples are the methods proposed by [17, 23-25, 130, 131].

2.4 State-of-the-art Local Invariant Feature Detectors

A few selected, representative local invariant feature detectors are described in more detail in this section as they are widely-used in computer vision systems today. The methods discussed are: SIFT [12], SURF [13], Harris-Laplace, Harris-Affine, Hessian-Laplace, Hessian-Affine [15], EBR [122], IBR [122], MSER [14], Salient Regions [18] and SFOP [16]. Although FAST [22, 23] is also quite popular, it has not been included here as it is not invariant to scale changes [1].

2.4.1 Scale Invariant Feature Transform (SIFT)

During the last decade or so, SIFT [12] has become the most popular technique for matching image features due to its fast detector coupled with a highly distinctive descriptor. The algorithm is divided into four main stages: scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor computation. The first two stages form

the detection phase, where in all scales and image locations are searched to identify potential interest points followed by 3-D quadratic interpolation to determine their location to sub-pixel, sub-scale accuracy. SIFT approximates Laplacian-of-Gaussian (LoG) with a Difference-of-Gaussians (DoG) filter to detect blobs in an image. This approximation leads to a speed gain in the feature detection phase of SIFT. Once detected, each keypoint location is assigned one or more orientations depending upon local image gradient directions. This is followed by descriptor computation: the algorithm employs a descriptor with 128 coefficients, based on the histogram of local oriented gradients around the interest point. The high dimensionality of SIFT descriptor directly affects the amount of computation required for feature matching and is considered a major shortcoming of SIFT for real-time applications [13].

2.4.2 Speeded-Up Robust Features (SURF)

The SURF algorithm [13] has two main (and distinct) stages, detection and description, followed by feature matching. SURF constructs a scale space by convolving rectangular masks of increasing size, corresponding to different scales, with the input image, using an integral image representation [132] for speed. This results in a series of blob response maps at different scales. The scale space is divided into a number of octaves, formed by grouping blob response maps for adjacent scales. Normally, four scales per octave are used as this is considered sufficient for scale space analysis [13]. The algorithm also doubles the spatial sampling interval with increasing octave to reduce computation. Once the scale-space is constructed, 3-D non-maximum suppression is performed [133], followed by 3-D quadratic interpolation [134], to achieve sub-pixel, sub-scale accuracy. A blob response threshold is normally applied to select high-contrast interest points. The descriptors for the detected interest points, based on sums of Haar wavelet responses, are calculated after orientation assignment, to achieve rotation invariance. The final stage is image feature matching on the basis of computed descriptors by employing a nearest neighbor matching scheme [12].

2.4.3 Harris-Laplace/Affine

The Harris-Laplace detector [15] is based on the multi-scale Harris corner detector and the Laplacian operator. The algorithm consists of two main stages: the utilization of the multi-scale Harris corner detector for determining the image location of the local features; and selection of the characteristic scale of a local structure, for which a given function attains an extremum over scales. The Laplacian operator is utilized for this purpose. At the selected scale, there is maximum similarity between the feature detection operator and the local image structures.

The Harris-Affine detector [15] is based on the Harris-Laplace detector. It iteratively estimates the elliptical affine regions to obtain affine invariant corners. The algorithm utilizes the features detected by the Harris-Laplace detector and estimates the affine shape with the second moment matrix. The affine region is then normalized to a circular one and the new location and scale in the normalized image are determined. In the event that eigenvalues of the second moment matrix for the new point are not equal, the affine shape is again estimated with the second moment matrix and the whole process is repeated until the eigenvalues become equal.

2.4.4 Hessian-Laplace/Affine

These detectors are usually categorized as blob detectors. The Hessian-Laplace detector [15] is based on similar principles that are used by the Harris-Laplace detector. The main difference is that it employs the determinant of the Hessian matrix instead of the multi-scale Harris corner detector for determining the image location of the local features. Like the Harris-Laplace detector, it utilizes the Laplacian operator for selecting the characteristic scale at which there is maximum similarity between the feature detection operator and the local image structures.

The Hessian-Affine detector [15] is similar in spirit to the Harris-Affine detector in the sense that it follows the same procedure for detecting

affine invariant regions. The only dissimilarity is in the utilization of the features detected by the Hessian-Laplace detector as initial regions, instead of the ones detected by the Harris-Laplace detector.

2.4.5 Edge-Based Regions (EBR)

Since edges are stable image features that can be detected over a range of viewpoints, scales and illumination changes, the EBR detector [122] utilizes the edges present in an image for detecting affine invariant regions. The algorithm consists of the following stages: selection of Harris corners [4]; detection of Canny edges [135]; evaluation of relative affine invariant parameter along edges; construction of one-dimensional family of parallelograms; and selection of a parallelogram based on local extrema of invariant function. EBR usually detects corner-like structures in an image and is considered to perform well on structured scenes due to its dependence on the presence of edges in the given image.

2.4.6 Intensity-Based Regions (IBR)

The IBR detector [122] finds affine invariant blob-like structures in an image by starting from intensity extrema detected at multiple scales and then exploring the image around them in a radial way, delineating regions of arbitrary shape and finally replacing them by ellipses. The main stages of the algorithm are: selection of intensity extrema; consideration of intensity profile along rays; selection of maximum of invariant function along each ray; connection of all local maxima to enclose an affine-invariant region; and ellipse fitting on the irregularly shaped region. Since the direct output of IBR can be any closed boundary of a segmented region [19], this algorithm is usually categorized as a segmentation-based detector.

2.4.7 Maximally Stable Extremal Regions (MSER)

The MSER detector [14] is an affine invariant method which is based on the concept of computing a watershed-like segmentation with varying thresholds. It selects such regions that remain stable over a range of

thresholds. The feature detected by the MSER detector, known as the Maximally Stable Extremal Region, is a connected component for an appropriately thresholded image, which is often a blob-like structure similar to the features detected by IBR. The set of extremal regions is closed under continuous transformation of image coordinates and monotonic transformation of image intensities. Although detection of these regions is related to thresholding, no global threshold is used. The detector tests all thresholds and evaluates the stability of the connected components. The MSER detector is considered to perform well on structured images with uniform regions separated by strong intensity changes.

2.4.8 Salient Regions

The Salient Regions detector, which is based on information theory, searches for *salient* features in an image, where saliency is described as local complexity or unpredictability. By utilizing the entropy of the probability distribution function of intensity values within a local image region and the *self-dissimilarity* in scale-space of the feature, this detector provides well-localized features. The detector consists of two main steps: the entropy of the probability distribution function is evaluated at each pixel over the three parameter family of ellipses centered on that pixel, and the set of entropy maxima over scale and the corresponding ellipse parameters are recorded, which are termed the candidate salient regions; for each of the candidate regions, the magnitude of the derivative of probability distribution function with respect to scale is computed, which in turn is used to calculate saliency; the candidate regions are then ranked over the entire image using their saliency and a specific number of top ranked regions are finally selected as salient regions.

2.4.9 Scale Invariant Feature Operator (SFOP)

The SFOP detector [16] is a multi-type detector as it extracts different types of complementary image features, such as circles, corners and blobs, simultaneously. This feature detection method is a scale-space extension of

the detector proposed in [9], generalizing it from junctions to all types of spiral features by incorporating the general spiral feature model of [136]. The SFOP detector is based on the concept of searching points where the consistency of image regions with respect to a spiral model is locally optimal. As a result, simultaneous detection and classification of image structures with complementary properties over scale-space as interpretable and identifiable subclasses is achieved. Due to the complementarity of its detected features, SFOP is a considered a good feature detector for camera calibration and object recognition tasks, particularly in the case of poorly textured scenes [16].

2.5 Hardware-Based Local Feature Detection

In recent years, researchers have started to focus on hardware-based systems for real-time detection of local invariant image features. This section provides an overview of the significant technological advances made in this field over the last few years.

Computer vision techniques are generally computation-intensive in nature and are mostly implemented in software. The local feature detection methods are no exception. Since real-time performance is desirable for many computer vision applications, such as target tracking and mobile robot navigation, it is usually difficult for software-only implementations to achieve the real-time goal due to the high computational complexity of these algorithms. Modern desktop computers that employ multiple processors clocking at GHz frequencies are, surprisingly, not generally well-suited to computation intensive, real-time computer vision algorithms due to the limited image processing capabilities of underlying hardware architecture. General-Purpose computation on Graphics Processing Units (GPGPU) is an emerging trend that utilizes high memory bandwidth and huge computational resources of graphics hardware to speed up many applications including image processing and video processing [137]. Graphics Processors, however, have high power consumption (usually tens

of watts) that makes them unsuitable for embedded vision systems with restricted power resources. Such systems usually employ commercial off-the-shelf embedded computers that do not guarantee real-time performance as they run at much lower clock frequencies and have restricted computational and power resources as compared to graphics processors. Lack of computer architectures capable of processing image and video data is thus, a major hurdle in real-time vision processing.

In order to achieve real-time performance for computer vision applications, especially on embedded processors, the inherent parallelism found in this class of algorithms can be used to great advantage. This implies the design of special-purpose parallel hardware structures capable of real-time operation for computer vision algorithms and applications.

Table 2-1: Performance of the state-of-the-art local invariant feature detection algorithms on modern desktop computers

Algorithm	Computation Time with Platform description
SIFT	600 ms for detection and description of interest points for an image size of 640 x 480 on Pentium III running at 700 MHz [138]
SURF	610 ms for detection and description of 1529 interest points for an image size of 800 x 640 on Pentium IV running at 3 GHz [13]
Harris-Laplace	7 sec for detection of 1438 interest points for an image size of 800 x 640 on Pentium II running at 500 MHz [15]
Hessian-Laplace	700 ms for detection of 1979 interest points for an image size of 800 x 640 on Pentium IV running at 3 GHz [13]
Harris-Affine	36 sec for detection of 1123 interest points for an image size of 800 x 640 on Pentium II running at 500 MHz [15]
Hessian-Affine	2.73 sec for detection of 1649 regions for an image size of 800 x 640 on Pentium IV Linux PC running at 2 GHz [46]
MSER	140 ms for detection of regions for an image size of 530 x 350 on a Linux PC with the Athlon XP 1600+ Processor [14]
Salient Regions	33 min and 33.89 sec for detection of 513 regions for an image size of 800 x 640 on Pentium IV Linux PC running at 2 GHz [46]
EBR	2 min and 44.59 sec for detection of 1265 regions for an image size of 800 x 640 on Pentium IV Linux PC running at 2 GHz [46]
IBR	10.82 sec for detection of 679 regions for an image size of 800 x 640 on Pentium IV Linux PC running at 2 GHz [46]

The performance of the state-of-the-art feature detection algorithms on modern desktop computers is far from real-time due to the high level of computational complexity involved. The execution times for software-only implementations of some popular feature detectors are given in Table 2-1. These clearly demonstrate the inability of desktop computers to run these algorithms in real-time. Special-purpose hardware architectures exploiting inherent parallelism of these algorithms are therefore required in order to achieve significant speed gain.

The work presented in [138] is considered ground breaking in the area of hardware-based local invariant feature detection. An FPGA based, fixed-point implementation of SIFT algorithm was targeted to achieve speed gain over software implementations. As a first step, a floating-point software implementation of the SIFT algorithm was converted to fixed-point, and modifications were made to routines so as to make them efficient for hardware implementation. Instead of using low-level hardware description languages like VHDL and Verilog, a high level tool known as System Generator was employed for major part of this particular hardware implementation. In this work, VHDL was used only for implementing low-level processes like DMA transfers and memory accesses, to make them more efficient. The final bit file for the FPGA was generated using Xilinx ISE. The Virtex-II Xilinx FPGA-based design reduced execution time of SIFT to 60 ms for an image size of 640 x 480 pixels, compared to 600 ms required on a Pentium-III 700 MHz processor. In [139], it is reported that this FPGA-based design is capable of computing SIFT features at a rate of 7 Hz for an image size of 1024 x 768.

A partial hardware implementation of the SIFT algorithm is described in [140] for online stereo calibration. Only two main components of the SIFT algorithm, i.e., Gaussian pyramid and Sobel filter, were implemented in Virtex-II Xilinx FPGA using VHDL, whereas the remaining ones were executed in software on a host computer. A pipelined hardware architecture clocking at 54 MHz was designed for implementing the Gaussian pyramid in a way that allowed feature extraction to start before

the image was fully digitized. A Sobel operator was also implemented in FPGA, instead of calculating edge gradient and orientation using finite differences. This architecture was capable of operating at 60 frames per second and reduced by 50–70% the time for feature extraction.

An innovative pipelined hardware architecture for Harris-Affine feature detector is presented in [141, 142], claiming to be the first attempt at implementation of a complex iterative algorithm in reprogrammable hardware. This fixed-point implementation was unique in a sense that it employed multiple FPGAs for extraction of scale- and rotation-invariant features. The coding was done in VHDL and was compiled using the Quartus-II software provided by Altera. The computation was distributed among four Altera Stratix S80 FPGAs that were able to process standard video (640 x 480 pixels) at 30 frames per second. This hardware architecture achieved a speed gain of 90-9000 times over an equivalent software implementation of the Harris-Affine feature detector, depending upon the language of implementation and the computing platform.

Another FPGA-based partial implementation of the SIFT algorithm is discussed in [143, 144]. A hardware-software co-design strategy was preferred over pure hardware implementation; the hardware–software partitioning was done in such a way that the detection phase of the algorithm was implemented in hardware whereas the description phase was targeted to run in software on a MicroBlaze processor. This architecture was realized on a Xilinx XUP-Virtex-II Pro board but was only capable of processing one octave for the SIFT algorithm. With MicroBlaze running at 100 MHz, it was claimed that this architecture required 0.8 ms for detection and description of key points for an image size of 320 x 240 pixels.

An FPGA based implementation of the Maximally Stable Extremal Region (MSER) detector is described in [145]. The designed architecture was implemented on a Xilinx XC2VP100 FPGA and achieved performance of 54 frames per second for an image size of 320 x 240 pixels without using any off-chip memory.

In [146], a dedicated processor for SIFT-based object recognition is proposed. This processor was based on Visual Image Processing memory and Network-on-Chip. Ten SIMD processing elements were also integrated into this processor architecture for exploiting data- and task-level parallelism of the SIFT algorithm. An important feature of this architecture was its low-power consumption. For an input image size of 320 x 240, this dedicated processor was able to achieve 10.1–15.9 frames per second for SIFT feature extraction at 200 MHz.

A parallel hardware architecture for SIFT is proposed in [147] which utilizes a hardware–software co-design strategy. Except descriptor computation, which ran in software on a NIOS-II soft core processor, all other steps of the SIFT algorithm were implemented in hardware. This is the most complete implementation of the SIFT algorithm to date and provided accurate results that were similar to software implementations. With a NIOS-II soft core processor running at 100 MHz, this architecture required 33ms to extract SIFT features for an image size of 320 x 240 pixels; thus, it could achieve performance of up to 30 frames per second.

Finally, researchers have also targeted the SURF algorithm for hardware acceleration. Some innovative hardware architectures for this particular feature detector are presented in [148-151].

2.6 Summary

After introducing the field of local invariant feature detection, this chapter has provided an overview of the local feature detectors proposed in the literature. The state-of-the-art feature detection techniques, such as SIFT and SURF, that are prevalent in most computer vision systems today are discussed in more detail. Finally, the chapter has provided a summary of the progress made in efficient hardware implementation of local feature detectors.

3 Improved Repeatability Measures

In theory, theory and practice are the same. In practice, they are not.

ALBERT EINSTEIN

Since local feature detection has been one of the most active research areas in computer vision during the last decade, a large number of detectors have been proposed. The interest in feature-based applications continues to grow and has thus rendered the task of characterizing the performance of various feature detection methods an important issue in vision research. The most frequently-employed metric in this regard is repeatability, essentially a theoretical measure. However, it has been observed that this does not necessarily mirror actual performance. In this chapter, after identifying the limitations of the original repeatability metric, improved repeatability measures are proposed which correlate much better with the true performance of feature detectors. Comparative results for several state-of-the-art feature detectors are presented using these measures which provide new insights into their behavior under various geometric and photometric transformations.

3.1 Introduction

Accuracy of observation is the equivalent of accuracy of thinking.

WALLACE STEVENS

Broadly speaking, every designed system aims to achieve some particular goals. For example, a robotic hand which is developed to catch a ball would strive to accomplish this objective. Performance measures are proposed by studying these goals in an effort to improve the effectiveness and reliability of the designed system. Their role is critical as an ineffective metric can potentially lead to a wrong research direction. Conversely, a carefully designed performance indicator helps to pinpoint areas of strength and weakness accurately, serving essentially as the primary step for directing the research effort in the right direction.

Performance measures are an integral part of contemporary local feature detection research and have played a vital part in the considerable progress in the field. For any feature detector, offline testing utilizing reliable measures (here the author means testing before deployment in the actual environment) is crucial to its success as the systems of which they are a component generally operate in complex and unknown environments across a wide spectrum of applications.

While designing performance indicators for local feature detectors, it is important to take into account the dependence of the feature description and feature matching stages on the selected features. If not considered properly, a feature detector may show good performance based on a specific measure when tested in isolation but may provide poor results in practical applications. A well-designed performance measure would gauge the performance not only of the feature detector but of the whole image matching system (see Figure 1-1).

Another useful aspect of performance metrics is the relative comparison of different feature detection techniques. Vision researchers today have a range of feature detectors available. However, there can be a

huge variation in results for a particular application depending on the detector used [1, 47]. It is therefore widely agreed that the performance characterization of feature detectors is an important issue in vision research [1, 46-48]. Performance metrics provide a systematic way of selecting a suitable feature detector for solving any particular vision problem under various geometric and photometric transformations. Comparison of competing algorithms based on an unreliable and ineffective metric can potentially lead to the selection of a feature detector that is not capable of solving a vision problem under expected imaging conditions.

Although there has been a lot of activity in the research community regarding performance measures and evaluation based on them, there is still a lack of reliable and effective performance metrics [1]. The most frequently employed measure for characterizing the performance of feature detectors is repeatability [1, 15, 46]. It is a theoretical measure which requires ground truth information for estimating the performance of any given detector. However, it has been reported that a high repeatability score does not guarantee good performance in practical situations [1]. More specifically, repeatability does provide information about the theoretical performance of detector but does not always mirror actual performance.

There is hence a strong motivation to design reliable and effective performance measures for local feature detectors. By *reliable* here the author means that the metrics must provide results which are consistent with the true performance in practical scenarios for a variety of detectors across a number of datasets. The word e*ffective* here implies that the metrics must also have some utility from a systems design perspective. To bridge the abovementioned research gap, this chapter first identifies the problems of the original repeatability metric [15, 46] and then presents improved repeatability measures which correlate much better with the true performance of feature detectors. By using Pearson's correlation coefficient, it will be shown that these improved measures are more reliable than the original repeatability metric across a wide range of local feature detectors utilizing well-established datasets [50]. Evaluation of eleven state-of-the-art

feature detectors based on the proposed measures will be carried out to identify the relative strengths and weaknesses of the algorithms under different geometric and photometric transformations using the widely-used Oxford datasets [50].

The remainder of the chapter is structured as follows: Section 3.2 describes the related work on performance measures and evaluation of local invariant feature detectors based on these metrics. After investigating the limitations of repeatability, improved repeatability measures are presented and the results based on them are verified by using Pearson's correlation coefficient across a wide range of feature detectors on well-established datasets [50] in Section 3.3. Comparative results for several state-of-the-art local invariant feature detectors based on the proposed measures under various geometric and photometric transformations utilizing the widely-used Oxford datasets [50] are presented in Section 3.4, providing new insights into the strengths and weaknesses of various detection techniques. A summary of the work described in this chapter is presented in Section 3.5.

3.2 Related Work

This section provides a review of the performance measures presented so far regarding local feature detectors. It also gives a summary of the related work on evaluation of feature detectors based on these metrics.

Corner detectors are evaluated based on chain coded curves by [84]. Visual inspection is used for performance characterization of detectors in [6]. An evaluation of feature detectors based on a quantitative measure of the quality of detected dominant points is performed in [152]. In [153], the localization accuracy of interest point detectors is utilized as a performance measure for comparing them using different planar projective invariants for which reference values are computed using scene measurements. Three interest point detectors are evaluated by utilizing a L-corner model in [117]. In the same spirit, theoretical analysis of L-corners with aperture angles in the range 0–180° is used for comparing feature detectors in [154]. Alignment

of the extracted points, accuracy of the 3D reconstruction, accuracy of the epipolar geometry and stability of the cross-ratio are used as criteria to measure the localization accuracy of a model-based L-corner detector by [118]. An approach similar to [153] is utilized in [155] for comparing feature detectors.

A metric based on visual inspection is employed in [156] for evaluating feature detectors. Canny's criteria [135], namely good detection, good localization and low response multiplicity are used as performance metrics for theoretical evaluation of edge detectors by [157]. In [158], structure from motion is used as a specific task to characterize performance. Edge detectors are compared for object recognition task by [159]. Human marked ground-truth is utilized in [160] for assessing the performance of edge detectors. Collinearity, intersection at a single point, parallelism and localization on an ellipse are used as criteria for performance characterization of detectors in [161]. In [162], an evaluation of the quality of detection is carried out based on a set of visual inspection criteria.

Repeatability and information content are utilized as performance metrics in [47]. These two measures are also used in [163] for evaluating feature detectors in the context of image retrieval. The definition of repeatability was refined by [15] and used for evaluating six state-of-the-art local feature detectors in [46]. Consistency of the number of corners and accuracy are employed as performance metrics in [164]. The same approach is used by [165] for performance characterization of corner detectors. For evaluating performance of detectors in [98, 166], the number of frames over which the corners are detected during tracking is used as a measure. An evaluation of local feature detectors on non-planar scenes is carried out in [167, 168].

For the specific task of matching 3D object features across viewpoints and lighting conditions, an assessment of the performance of feature detectors is done by [169, 170]. Feature detectors are compared for recognition task using object category training data in [171, 172]. Clustering properties and compactness of feature clusters are employed as performance

metrics in [171]. For a pedestrian detection task, an evaluation of feature detectors is done in [173].

More recently, in the context of automatic image orientation systems, the performance characterization of local features is carried out by [45]. Localization accuracy of feature detectors is evaluated in [174]. A similar approach based on localization accuracy is reported in [175]. Completeness of detected features is used as a performance metric in [19, 176] for comparing state-of-the-art local feature detectors. In [177], the performance of detectors is evaluated under viewpoint, scale and light changes by using a large database of images with recall rate as performance measure.

Finally, the author would like to state that the literature on performance metrics and evaluation of local feature detectors based on them is vast and has grown rapidly after the emergence of SIFT [11, 12]. There have been a number of evaluations based on specific vision tasks such as visual SLAM [178] and face detection [179]. It is not possible to describe every such contribution here but an attempt has been made to mention all those developments which are considered important in this domain.

3.3 Improved Repeatability Measures

In this section, following a brief overview of the original repeatability metric [15, 46], its limitations are highlighted and then alternatives are proposed, with supporting results, which indicate the effect of various image transformations reliably and are more consistent with the actual performance of detectors. Here, actual performance means the true matches obtained using ground-truth homography after descriptor-based matching of detected points. A true match occurs when a nearest neighbor matched feature in one image is projected into another image using the ground-truth homography and its projection lies within 1.5 pixels of corresponding nearest neighbor matched feature in the other image. Any nearest neighbor matches that do not satisfy this criterion are classified as false matches (i.e., false alarms). As already mentioned in Section 1.1, the SIFT descriptor [11,

12] is utilized as it is considered to provide the best matching results among the available descriptors [55, 180]. For matching, the Nearest Neighbor matching scheme proposed by [12] is used.

3.3.1 An Overview of the Repeatability Metric

The evaluation of the performances of local feature detectors under various geometric and photometric transformations has become important, in order to identify their strengths and shortcomings for a range of vision applications. As is clear from Section 3.2, several approaches have been used in this regard including ground-truth verification, localization accuracy, theoretical analysis and specific tasks; however, the most widely employed measure for the performance characterization of feature detectors is the repeatability rate [1]. This metric was originally proposed by [47] and later refined by [15, 46]. In [47], the repeatability rate is defined as the ratio of the number of points repeated in the overlapping region of two images to the total number of detected points. An interest point is considered 'repeated' if its 2-D projection in the other image using planar homography lies within a neighborhood of size ε of an interest point detected in the other image. Since these feature detectors identify interest points at different scales, measuring the 2-D distance between interest points detected at different scales, to decide whether they are repeatable or not, may lead to inaccurate results. A more sophisticated definition of repeatability is presented in [15, 46], which also considers the overlap of scale-dependent regions centered in the interest points. This may be written mathematically as in [15, 46]:

$$Repeatability = \frac{\text{Total number of repeated points}}{\min(\text{points detected in image 1}, \text{points detected in image 2})} \qquad \textit{Equation 3-1}$$

3.3.2 Limitations of Repeatability

Despite being popular, it has been remarked that "repeatability does not guarantee high performance" [1]. The author has investigated the

repeatability metric in detail across a wide range of detectors by utilizing well-established datasets [50] and identified its shortcomings. Here, two sample cases are discussed to illustrate the problems with this metric and then the conclusions drawn from this investigation are presented.

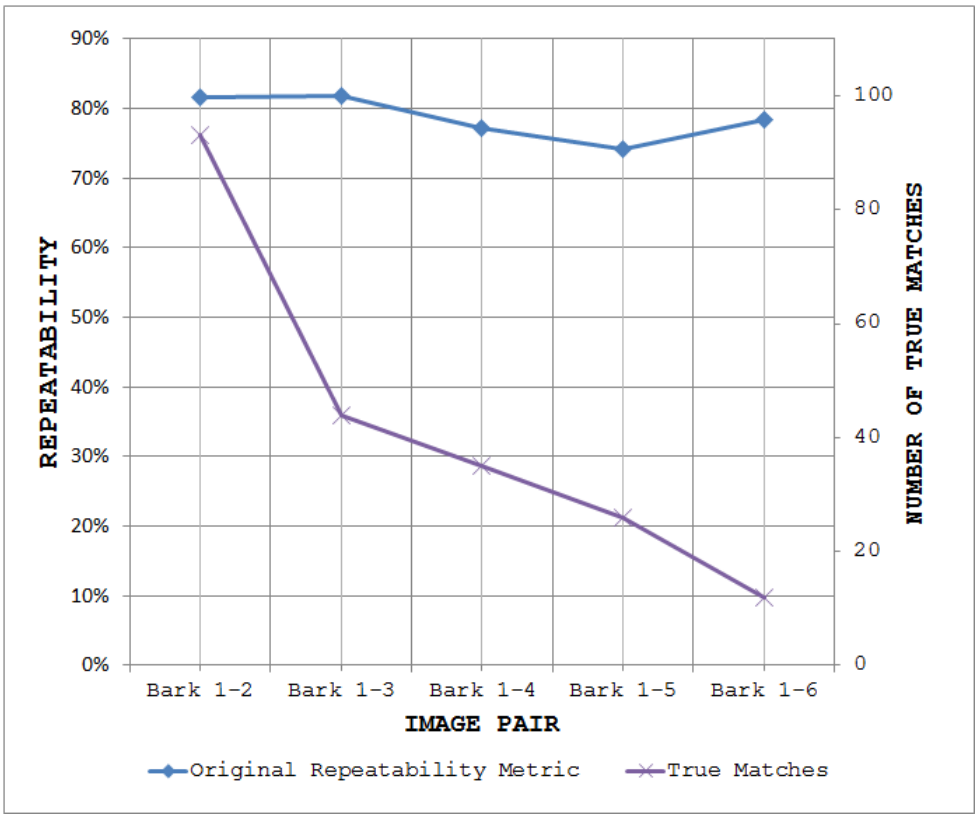

Figure 3-1: Repeatability curve and number of true matches for Hessian-Laplace detector with Bark dataset using the original metric

Figure 3-1 depicts the first sample case, the repeatability values and the numbers of true matches obtained for the Bark dataset [50] with the Hessian-Laplace detector utilizing the original metric [15, 46]. Note that repeatability values should be read from the left ordinate axis and the number of true matches from the right ordinate axis. The number of true matches was calculated for every image pair using ground-truth homography after SIFT descriptor based matching of the detected points. Observing the trends of the two curves (repeatability and number of true matches), it becomes evident that the original metric reports good performance, with only slight variation in repeatability values, but there is

a continuous decline in the actual performance of the detector as indicated by the decreasing number of true matches. Interestingly, for the image pair Bark 1 and Bark 6, the original repeatability metric shows that there is an improvement in performance with respect to the previous image pair (Bark 1 and Bark 5). This illustrates that the original repeatability metric [15, 46] overestimates the performance of the detector and fails to capture the effect of image transformation in its performance.

The second sample case is shown in Figure 3-2, the repeatability values and the numbers of true matches obtained for the Boat dataset [50] with SURF detector utilizing the original metric [15, 46]. As in Figure 3-1, note that repeatability values should be read from the left ordinate axis and the number of true matches from the right ordinate axis. Again, it is clear that the repeatability curve does not have high correlation with the true performance of SURF in practical situations. The curve for the number of true matches shows a continuous decay but the original repeatability metric reports an improvement in performance for the image pair Boat 1 and Boat 3 relative to the previous image pair (Boat 1 and Boat 2). This essentially means that the performance of SURF increases with the increasing amount of image transformation (zoom and rotation changes in this particular case) which is misleading.

From this investigation of the original repeatability metric [15, 46], the following limitations are identified:

1) The repeatability rate only partially reflects the effect of various geometric and photometric transformations as it considers the *minimum* number of interest points detected in either of the two images.
2) It is not always possible to predict the effect of a specific transformation on the number of corresponding points from the value of repeatability.
3) The reference image is not fixed when evaluating the performance of a detector for a specific dataset.

4) Repeatability does not always reflect the effect of transformation on the number of true matched points, *i.e.* the true performance.

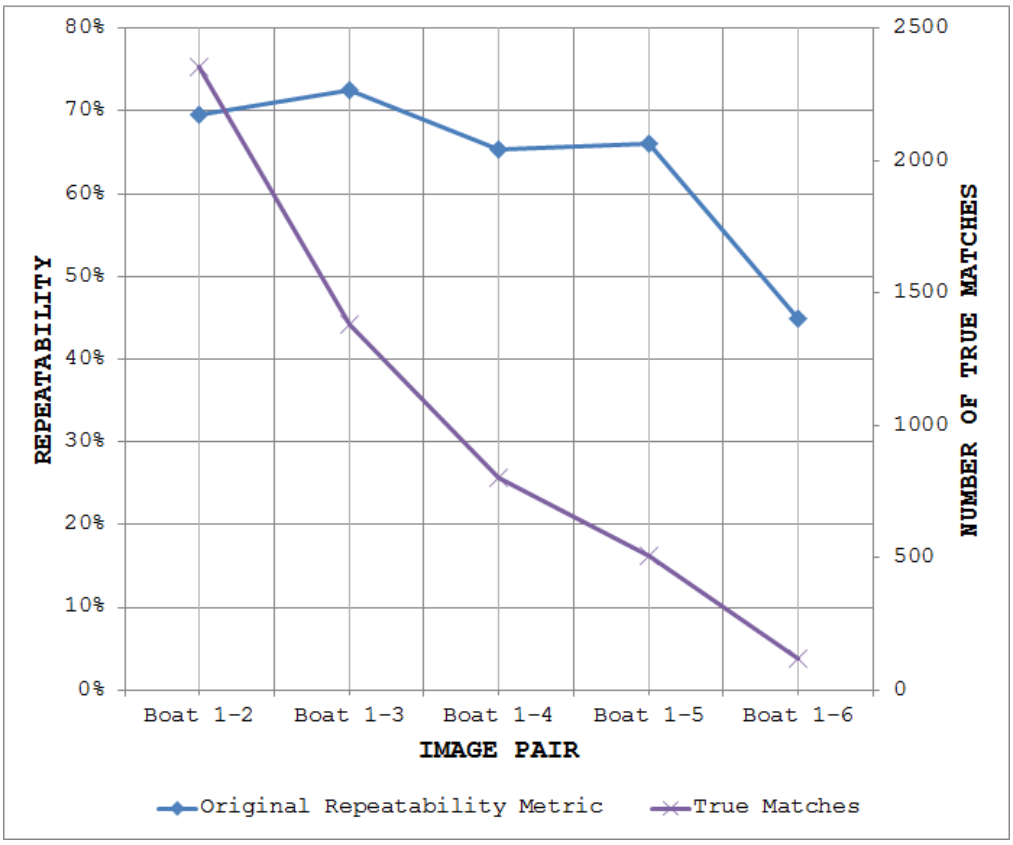

Figure 3-2: Repeatability curve and number of true matches for SURF detector with Boat dataset using the original metric

3.3.3 Proposed Measure 1

To overcome the above-mentioned shortcomings, two alternative definitions of repeatability are presented that are more consistent with the actual performance of feature detectors. Both use the same scale-dependent regions as [15, 46]. The first of these is appropriate for applications that involve image sequences, while the second is more suited to applications involving pairs of images (*e.g.,* computational stereo).

Unlike the definition in [15, 46], the sequence of images is not ignored when determining the effect of various photometric and geometric transformations; the first image in the sequence is considered as the 'reference' in all cases. The author also takes into account only those interest points that lie in the common part of the two images and defines an

interest point as 'repeatable' if ε < 1.5 pixels and the overlap error between scale-dependent regions centered in the two interest points, defined as:

$$Overlap\ error = 1 - \frac{\mu_a \cap (A^T \mu_b A)}{(\mu_a \cup A^T \mu_b A)} \qquad \textit{Equation 3-2}$$

is less than 40%, as in [15, 46], where μ_a and μ_b are the regions defined by $x^T \mu x = 1$ and A is the homography between the two images. The numerator of the fractional part in Equation 3-2 represents the intersection whereas the denominator represents the union of these regions. However, as opposed to [15, 46], which uses the minimum of the number of interest points detected in the two images, the repeatability rate is defined as:

$$Measure\ 1 = \frac{N_{rep}}{N_{ref}} \qquad \textit{Equation 3-3}$$

where N_{rep} is the total number of repeated points and N_{ref} is the total number of interest points in the common part of the reference image.

3.3.4 Proposed Measure 2

This measure follows the same framework as described above but employs a symmetric approach for the computation of repeatability rate:

$$Measure\ 2 = \frac{2 \times N_{rep}}{N_{ref} + N_{test}} \qquad \textit{Equation 3-4}$$

where N_{rep} is the number of repeated interest points, N_{ref} and N_{test} are the number of interest points detected in the common part of the scene in the reference and test images respectively.

3.3.5 Qualitative Results

To demonstrate the utility of the proposed measures, the two sample cases of Section 3.3.2 are used again.

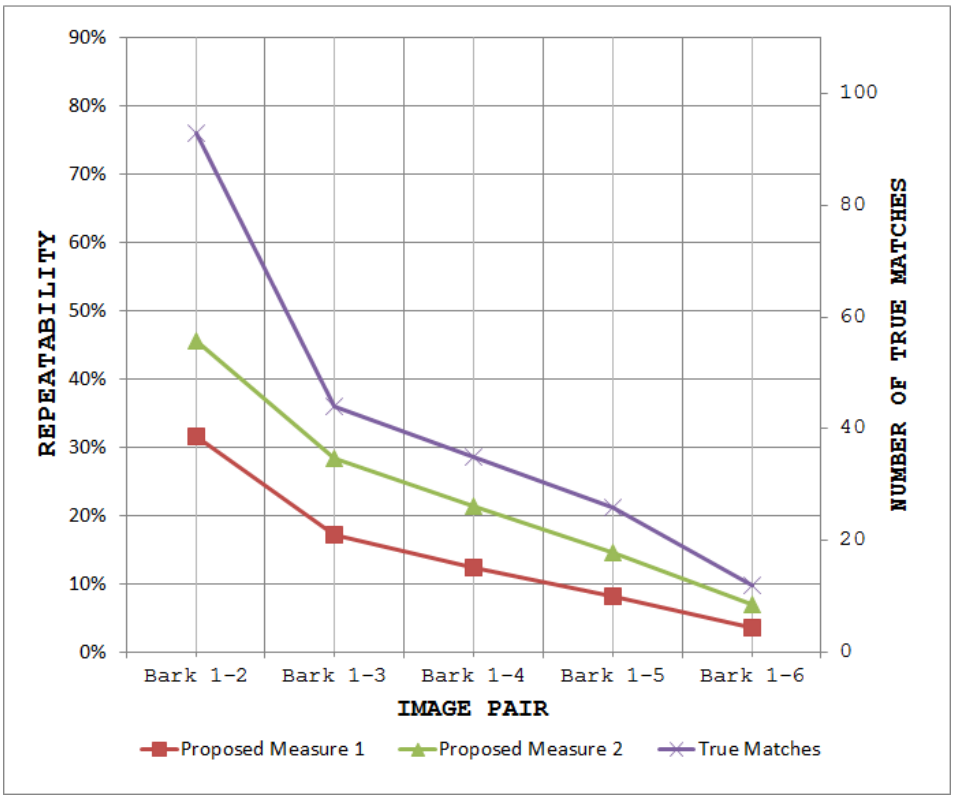

Figure 3-3: Repeatability curves and number of true matches for Hessian-Laplace detector with Bark dataset using the improved measures

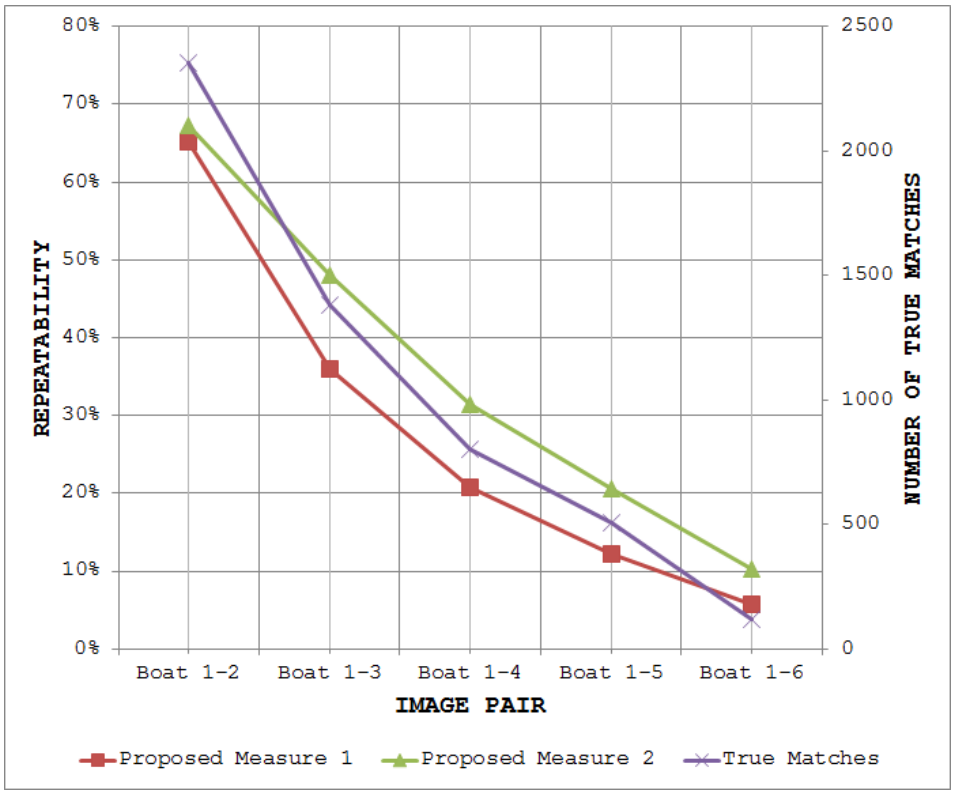

Figure 3-4: Repeatability curves and number of true matches for SURF detector with Boat dataset using the improved measures

Figure 3-3 shows the repeatability values and the numbers of true matches obtained for the Bark dataset with the Hessian-Laplace detector utilizing the proposed measures. As compared to the original repeatability metric (see Figure 3-1), the improved measures achieve results which are much better correlated with the true performance of Hessian-Laplace. The results for the second sample case (SURF with Boat dataset) utilizing the proposed measures are depicted in Figure 3-4. The same trend shown by all three curves (number of true matches and repeatability curves) in this figure highlights the usefulness of the improved repeatability measures.

3.3.6 Verification of Improved Measures using Pearson's Correlation Coefficient

For the proposed measures to be reliable and have any value, the performance results obtained utilizing them must be consistent with the true performances for a variety of feature detectors across a number of datasets. Here, results are presented that verify the accuracy and reliability of the proposed repeatability metrics.

Repeatability values were computed for the widely-used Oxford datasets [50] using the original repeatability metric [15, 46] and the two proposed measures. Results were obtained for eleven state-of-the-art feature detectors, namely SIFT, SURF, Harris-Laplace, Hessian-Laplace, Harris-Affine, Hessian-Affine, MSER, IBR, EBR, Salient and SFOP, using their original implementations with default parameters [1, 12-16, 18, 122]. For all detectors, the number of true matches was also calculated for every image pair using the ground-truth homography after SIFT descriptor based matching of detected points. To measure how well the three calculated repeatability curves agree with the number of true matches, Pearson's correlation coefficient, r, is used. Correlation coefficient values with corresponding p-values for the above mentioned detectors are given in Table 3-1–Table 3-11; note that a p-value gives the probability that the corresponding correlation value is *incorrect*. A discussion of these results is given in the next few paragraphs.

SURF. Table 3-1 presents the results for the SURF detector. For the original repeatability metric, the correlation coefficient values indicate that the results are highly correlated with the actual performance of SURF for only the Bark, Graffiti, UBC and Wall datasets. Although it is clear that the results are not particularly consistent with true performances for the remaining four datasets, the case of Trees dataset is noteworthy and deserves more discussion. The negative value of the correlation coefficient here means that the original repeatability metric is reporting a performance score which is *opposite* to the actual performance of SURF (i.e., the true performance is decreasing whereas the original repeatability metric is showing an increase in performance). This is certainly not desirable and indicates the unreliability of the original repeatability metric. On the other hand, it is evident that the results obtained by utilizing the improved repeatability measures are highly correlated with the actual performance of the detector.

SIFT. Presented in Table 3-2 are the results for the SIFT detector. Again, for the improved repeatability measures, the results are consistent with the actual performances for all datasets as indicated by large positive values of the correlation coefficient. The unreliability of the original repeatability metric is again highlighted by the negative value of correlation coefficient for the Bark dataset. It should be noted that, for the same dataset, the results achieved utilizing the improved repeatability measures manage to mirror actual performances reasonably well.

Harris-Laplace. Table 3-3 illustrates the results for the Harris-Laplace detector. Except for the Bark dataset with the original repeatability metric, all other results obtained using the three metrics indicate high degrees of correlation with the actual performances of the Harris-Laplace detector.

Hessian-Laplace. It is evident from Table 3-4 that the results obtained using the improved repeatability measures are consistent with the true performance of the Hessian-Laplace detector for all datasets. On the other

hand, the original repeatability metric fails to achieve large values of correlation coefficient for the Bark, Boat and Trees datasets.

Table 3-1: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for SURF detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.884	0.0466	0.997	0.0002	0.989	0.0014
Bikes	0.768	0.1287	0.996	0.0003	0.985	0.0022
Boat	0.697	0.1909	0.996	0.0003	0.993	0.0007
Graffiti	0.939	0.0179	0.971	0.0059	0.960	0.0095
Leuven	0.778	0.1213	0.998	0.0001	0.991	0.001
Trees	-0.591	0.2940	0.991	0.001	0.968	0.0068
UBC	0.990	0.0012	0.998	0.0001	0.999	0.000
Wall	0.889	0.0436	0.950	0.0133	0.929	0.0225

Table 3-2: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for SIFT detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	-0.301	0.6226	0.853	0.0661	0.884	0.0466
Bikes	0.962	0.0088	0.995	0.0004	0.999	0.000
Boat	0.963	0.0085	0.989	0.0014	0.997	0.0002
Graffiti	0.948	0.0141	0.973	0.0053	0.964	0.0082
Leuven	0.860	0.0615	0.993	0.0007	0.978	0.0039
Trees	0.942	0.0166	0.969	0.0065	0.959	0.0099
UBC	0.918	0.0278	0.865	0.0583	0.876	0.0514
Wall	0.891	0.0425	0.944	0.0158	0.908	0.0330

Table 3-3: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for Harris-Laplace detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.686	0.201	0.980	0.0034	0.983	0.0027
Bikes	0.984	0.0024	0.983	0.0027	0.983	0.0027
Boat	0.918	0.0278	0.992	0.0009	0.998	0.0001
Graffiti	0.889	0.0436	0.938	0.0184	0.919	0.0273
Leuven	0.962	0.0088	0.989	0.0014	0.980	0.0034
Trees	0.820	0.0892	0.925	0.0244	0.894	0.0408
UBC	0.994	0.0006	0.996	0.0003	0.987	0.0018
Wall	0.885	0.0460	0.959	0.0099	0.934	0.0202

Harris-Affine. Table 3-5 shows the results for the Harris-Affine detector. All three metrics achieve good results as indicated by large positive values of correlation coefficient.

Hessian-Affine. Given in Table 3-6 are the results for the Hessian-Affine detector. The improved repeatability measures again yield results that have high correlation with the actual performance of Hessian-Affine detector. Barring the Boat dataset, the original repeatability also achieves good results.

SFOP. Table 3-7 presents the results for the SFOP detector. For all datasets, the results obtained utilizing the original repeatability metric and the improved measures are consistent with the true performance of SFOP.

Table 3-4: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for Hessian-Laplace detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.628	0.2566	0.994	0.0006	0.984	0.0024
Bikes	0.954	0.0118	0.994	0.0006	0.983	0.0027
Boat	0.742	0.1511	0.998	0.0001	0.996	0.0003
Graffiti	0.866	0.0577	0.950	0.0133	0.921	0.0263
Leuven	0.936	0.0192	0.999	0.0000	0.993	0.0007
Trees	0.792	0.1103	0.881	0.1197	0.922	0.0258
UBC	0.996	0.0003	0.996	0.0003	0.997	0.0002
Wall	0.919	0.0273	0.971	0.0059	0.953	0.0121

Table 3-5: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for Harris-Affine detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.918	0.0278	0.953	0.0121	0.971	0.0059
Bikes	0.978	0.0039	0.998	0.0001	0.993	0.0007
Boat	0.889	0.0436	0.995	0.0004	0.997	0.0002
Graffiti	0.968	0.0068	0.992	0.0009	0.983	0.0027
Leuven	0.989	0.0014	0.998	0.0001	0.993	0.0007
Trees	0.941	0.0171	0.943	0.0162	0.928	0.0229
UBC	0.996	0.0003	0.996	0.0003	0.994	0.0006
Wall	0.950	0.0133	0.986	0.0020	0.975	0.0047

Table 3-6: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for Hessian-Affine detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.887	0.0448	0.986	0.0020	0.990	0.0012
Bikes	0.972	0.0056	0.993	0.0007	0.985	0.0022
Boat	0.788	0.1134	0.999	0.0000	0.993	0.0007
Graffiti	0.943	0.0162	0.986	0.0020	0.970	0.0062
Leuven	0.973	0.0053	0.998	0.0001	0.996	0.0003
Trees	0.851	0.0675	0.838	0.0763	0.938	0.0184
UBC	0.998	0.0001	0.998	0.0001	0.998	0.0001
Wall	0.969	0.0065	0.994	0.0006	0.987	0.0018

Salient. In Table 3-8 are provided the results for Salient detector. The results clearly indicate the high reliability and accuracy of the proposed measures. On the other hand, the original repeatability metric achieves a poor value of correlation coefficient for the Leuven dataset which highlights its failure to describe the actual performance of Salient detector.

MSER. High degree of correlation of the results achieved utilizing the proposed measures with the true performance of MSER detector for all datasets in Table 3-9 provides evidence to their dependability. The original repeatability metric fails to achieve high correlation for the Leuven and Boat datasets.

IBR. Illustrated in Table 3-10 are the results for IBR. It can be seen clearly that the proposed repeatability measures reflect the true performance of IBR for all datasets. In the case of Bark and Leuven datasets, the original repeatability metric accomplishes a relatively low value of correlation coefficient.

EBR. Finally, Table 3-11 presents the results for EBR. For the original repeatability metric, the results for Boat and Trees again fail to achieve high degree of correlation with the true performance. However, the worst case of all is the Bikes dataset, for which the original repeatability metric provides a negative correlation coefficient value. As mentioned before, this is

undesirable and indicates the inconsistency of the metric. Conversely, the improved repeatability measures display a reliable behavior once more for all datasets.

Table 3-7: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for SFOP detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.839	0.0751	0.943	0.0160	0.962	0.0086
Bikes	0.985	0.0021	0.993	0.0006	0.988	0.0014
Boat	0.869	0.0553	0.998	0.0000	0.992	0.0008
Graffiti	0.951	0.0127	0.979	0.0035	0.969	0.0064
Leuven	0.994	0.0005	0.998	0.0001	0.996	0.0002
Trees	0.983	0.0026	0.877	0.0503	0.976	0.0043
UBC	0.991	0.0010	0.984	0.0024	0.994	0.0005
Wall	0.912	0.0308	0.950	0.0130	0.935	0.0193

Table 3-8: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for Salient Regions detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.871	0.0544	0.943	0.0162	0.960	0.0094
Bikes	0.965	0.0077	0.991	0.0009	0.984	0.0024
Boat	0.957	0.0106	0.997	0.0001	0.997	0.0002
Graffiti	0.963	0.0082	0.983	0.0027	0.976	0.0044
Leuven	0.523	0.3656	0.989	0.0012	0.975	0.0047
Trees	0.874	0.0523	0.995	0.0003	0.961	0.0090
UBC	0.935	0.0195	0.956	0.0107	0.934	0.0199
Wall	0.907	0.0331	0.959	0.0097	0.943	0.0162

Table 3-9: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for MSER detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.989	0.0012	0.993	0.0006	0.992	0.0007
Bikes	0.961	0.0090	0.997	0.0001	0.989	0.0012
Boat	0.780	0.1194	0.999	0.0000	0.990	0.0011
Graffiti	0.971	0.0056	0.996	0.0003	0.993	0.0007
Leuven	0.467	0.4274	0.996	0.0002	0.986	0.0020
Trees	0.994	0.0004	0.984	0.0024	0.970	0.0060
UBC	0.997	0.0001	0.997	0.0001	0.981	0.0030
Wall	0.978	0.0037	0.995	0.0003	0.990	0.0010

General Conclusions. These results demonstrate the high reliability of repeatability values obtained using the proposed measures. For all combinations of the eleven state-of-the-art feature detectors and eight datasets [50], the mean value of the correlation coefficient is: 0.850 with standard deviation 0.286 for the original repeatability metric, 0.977 ± 0.033 for proposed measure 1, and 0.973 ± 0.027 for proposed measure 2.

Table 3-10: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for IBR detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.751	0.1433	0.996	0.0003	0.977	0.0041
Bikes	0.958	0.0102	0.999	0.0000	0.996	0.0003
Boat	0.839	0.0754	0.999	0.0000	0.986	0.0020
Graffiti	0.983	0.0026	0.998	0.0001	0.995	0.0003
Leuven	0.761	0.1348	0.999	0.0000	0.997	0.0002
Trees	0.996	0.0002	0.987	0.0017	0.967	0.0070
UBC	0.996	0.0002	0.999	0.0000	0.998	0.0001
Wall	0.988	0.0015	0.998	0.0001	0.995	0.0003

Table 3-11: Pearson's correlation coefficients and corresponding p-values calculated using repeatability values and true matches for EBR detector

Datasets	Original Metric		Measure 1		Measure 2	
	r	p-value	r	p-value	r	p-value
Bark	0.862	0.0600	0.957	0.0103	0.936	0.0190
Bikes	-0.782	0.1175	0.998	0.0000	0.993	0.0007
Boat	0.786	0.1145	0.997	0.0001	0.975	0.0045
Graffiti	0.922	0.0258	0.997	0.0001	0.985	0.0021
Leuven	0.965	0.0077	0.995	0.0003	0.988	0.0015
Trees	0.732	0.1592	0.981	0.0030	0.957	0.0105
UBC	0.994	0.0005	0.955	0.0111	0.994	0.0005
Wall	0.939	0.0175	0.992	0.0008	0.980	0.0032

3.4 Evaluation of State-of-the-art Detectors

Having established the authenticity and reliability of the proposed measures in the previous section, these improved metrics are now employed to perform a relative performance comparison of the eleven state-of-the-art feature detectors mentioned in Section 3.3.6 under various geometric and photometric transformations. The widely-used Oxford datasets [50] and

authors' original programs (binary or source) are utilized with parameters set to values recommended by them in an effort to make these results a direct complement to existing evaluations.

3.4.1 Results under Various Transformations using Proposed Measure 1

Figure 3-5 to Figure 3-12 depict the comparative results for the state-of-the-art feature detectors under various image transformations utilizing proposed measure 1. Since the evaluation work done by [46] is considered the most comprehensive in this domain, the similarities and contradictions of the author's findings with those reported in that study are also discussed. The following six feature detectors were considered in [46]: Harris-Affine, Hessian-Affine, MSER, EBR, IBR and Salient. The author however is investigating a larger group of feature detectors to encompass the recent advancements in the field, such as SURF [13] and SFOP [16]. A discussion of these results is given below.

Bark dataset. This dataset involves zoom and rotation changes for a textured scene. Figure 3-5 shows the results for this dataset. It is evident that the repeatability scores for all the detectors under investigation decrease with the increasing amount of transformation. This is contradictory to the results presented in [46] which show that the performance of most detectors, especially Hessian-Affine, Harris-Affine and MSER, is little affected in the case of Bark dataset. Moreover, the repeatability scores for all detectors are much lower than those presented in [46]. For example, Hessian-Affine, the detector which is declared the best for this particular dataset is shown to achieve a repeatability score of around 80% for the image pair Bark 1 and Bark 2 in [46], whereas according to the proposed measure, its value is about 32% (see Figure 3-5). The author also finds that the repeatability curve of Salient is continuously decaying—another discrepancy from the results presented in [46] which show a highly unstable behavior of Salient detector. According to Figure 3-5, SIFT and

Salient seem to be the best detectors for increasing amounts of zoom and rotation. EBR shows the worst performance of all the detectors, in agreement with the results shown for EBR in [46]. Harris-Affine and MSER also perform poorly, contradictory to the previous findings [46].

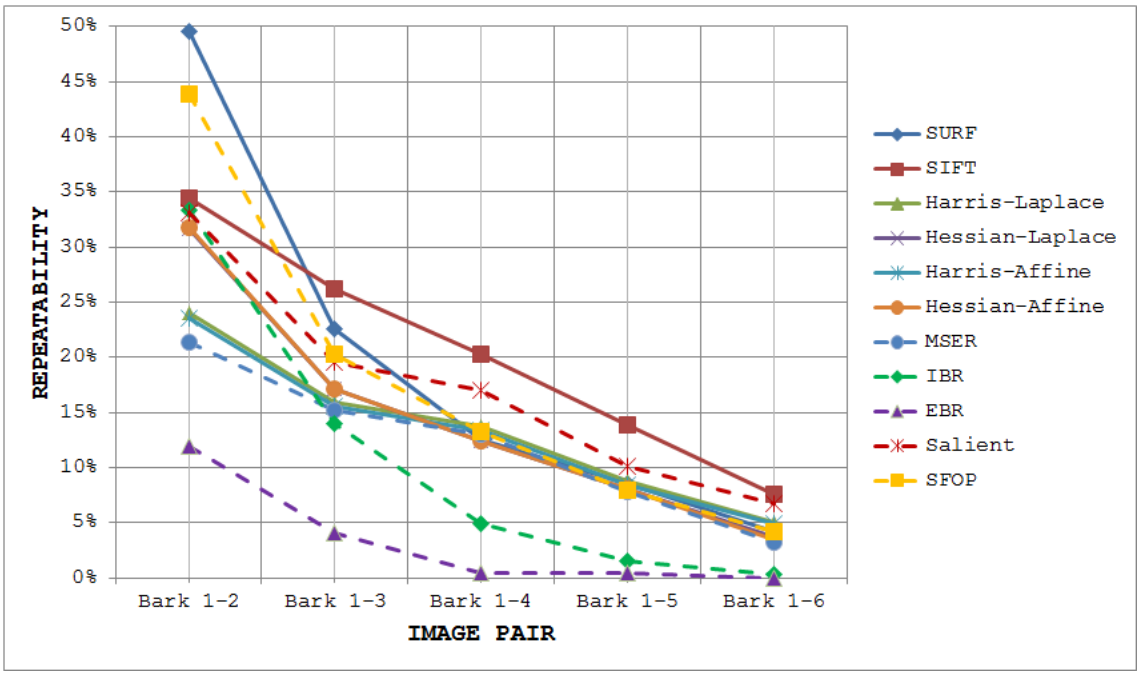

Figure 3-5: Repeatability results for state-of-the-art detectors for Bark dataset (zoom and rotation) using proposed measure 1

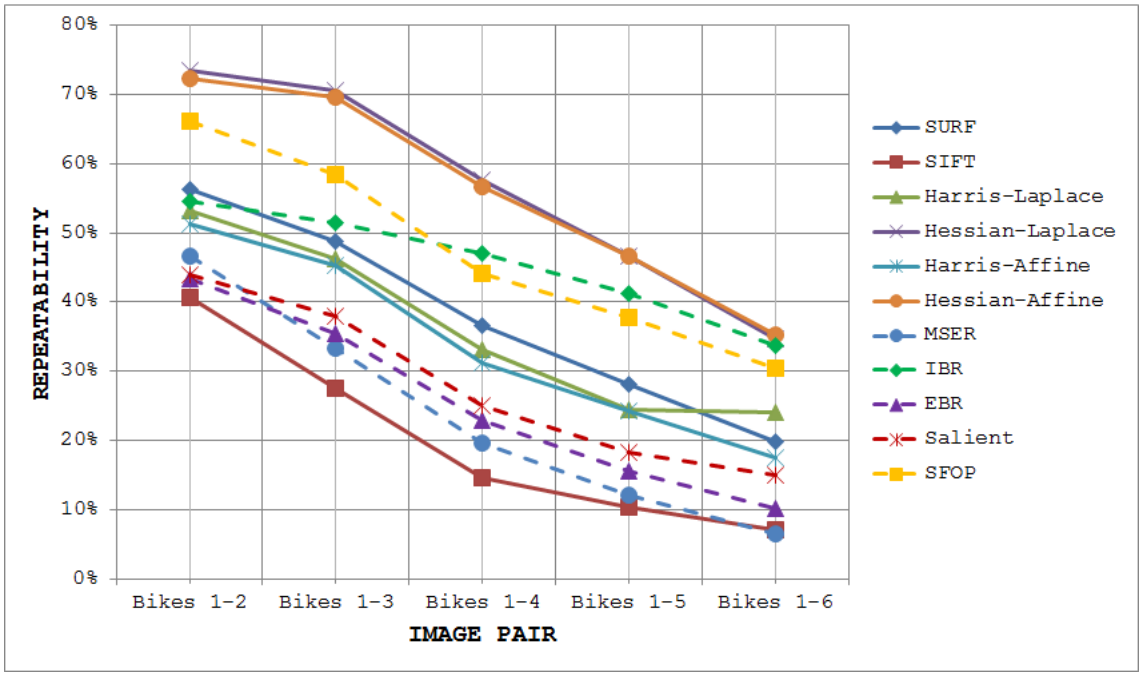

Figure 3-6: Repeatability results for state-of-the-art detectors for Bikes dataset (blur) using proposed measure 1

Bikes dataset. Figure 3-6 shows the results for the Bikes dataset, which are fundamentally different from [46]. This particular dataset involves increasing amounts of blur for a structured scene. In [46], it is claimed that all detectors investigated in that study show high invariance to image blur resulting in nearly horizontal repeatability curves except for MSER. This is totally contradictory to the presented results, which demonstrate a continuous decrease in performance for all the detectors with increasing image blur, a closer agreement with intuition. Hessian-Laplace and Hessian-Affine achieve the best performances for this particular dataset (see Figure 3-6). Although Hessian-Affine is also ranked as the top detector for the Bikes dataset in [46], the author's results show that its performance is significantly affected by the increasing amount of blur, which disagrees with the almost horizontal repeatability curve for Hessian-Affine shown in [46]. From Figure 3-6, it is also evident that SFOP and IBR demonstrate good performance for the Bikes dataset—again a deviation from the findings of [46], which ranks IBR quite low. The most notable result, however, is for the SIFT detector, which is outperformed by all other detectors. EBR, which is shown as one of the top two detectors for this dataset in [46] and essentially appears to improve its performance with the increasing amount of blur according to those results, also performs poorly as indicated by the low repeatability scores in Figure 3-6.

Boat dataset. Presented in Figure 3-7 are the comparative results of state-of-the-art detectors for the Boat dataset, a structured scene with zoom and rotation changes. Observing the repeatability curves shown in Figure 3-7, it appears that nearly all detectors have similar degradation of performance with the increasing amount of image transformation. Again, this loss in performance is substantially larger than the results presented in [46]. Hessian-Laplace and Hessian-Affine seem marginally better than the other detectors. While Hessian-Laplace is also identified as the best performer with a much flatter repeatability curve in [46], MSER achieves moderate repeatability scores in this investigation, which is contradictory to [46].

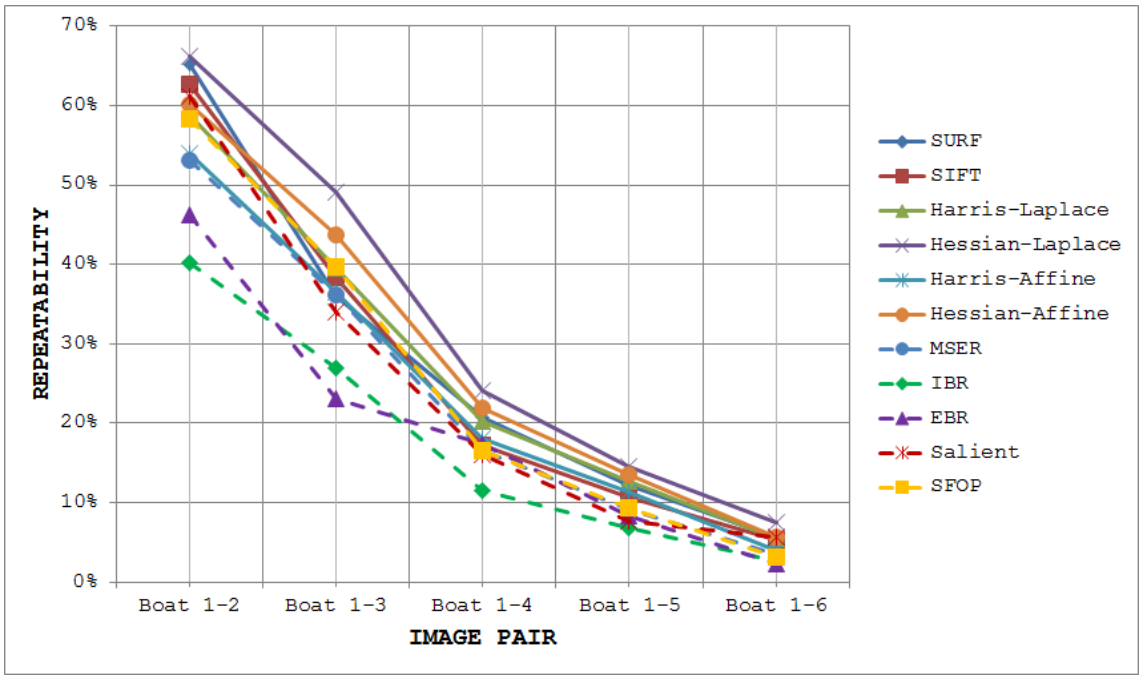

Figure 3-7: Repeatability results for state-of-the-art detectors for Boat dataset (zoom and rotation) using proposed measure 1

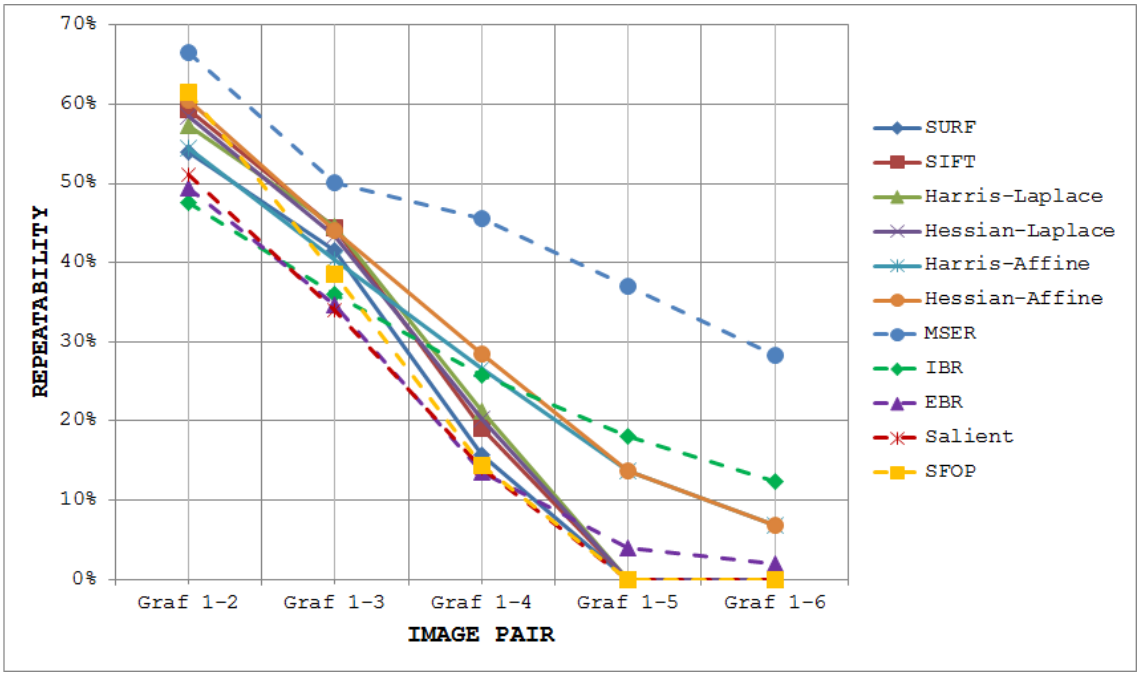

Figure 3-8: Repeatability results for state-of-the-art detectors for Graffiti dataset (viewpoint) using proposed measure 1

Graffiti dataset. This dataset consists of a structured scene with increasing viewpoint changes. Figure 3-8 depicts the performance of the detectors utilizing proposed measure 1. It is can be seen clearly that MSER comprehensively out-performs all other detectors, in agreement with [46]. On the other hand, Hessian-Affine and Harris-Affine show similar performances in this study which contradicts [46]. From Figure 3-8, it is evident that all other detectors barring Hessian-Affine, Harris-Affine, EBR, MSER and IBR, fail under increasing image transformation. Salient, which is shown to have reasonable repeatability for large viewpoint changes in [46], performs poorly for the last two image pairs (Graffiti 1 and Graffiti 5; Graffiti 1 and Graffiti 6) according to the presented results.

Leuven dataset. Figure 3-9 demonstrates the adverse effect of uniform light changes on the performance of the state-of-the-art detectors. Again, the presented results largely disagree with the findings of [46]. A substantial decline in performance is noticed with decreasing illumination for all the detectors in Figure 3-9, whereas it is concluded in [46] that the detectors under study have good robustness to illumination changes and achieve nearly horizontal repeatability curves. SFOP, SIFT and SURF seem to be the best detectors for the Leuven dataset (see Figure 3-9). MSER, Hessian-Affine and Harris-Affine show relatively low repeatability scores according to the presented results—a contradiction once again as they are shown as the top three detectors in [46].

Trees dataset. This particular dataset involves increasing image blur for a textured scene. The results for this dataset are presented in Figure 3-10. Hessian-Laplace, SURF, Hessian-Affine and SFOP show good performances under increasing image blur. SIFT appears to be severely affected by this type of image transformation. MSER also shows low repeatability scores which disagree with [46]. The performance of EBR is poor even for the images which do not have large amounts of blurring: this largely agrees with the results of [46].

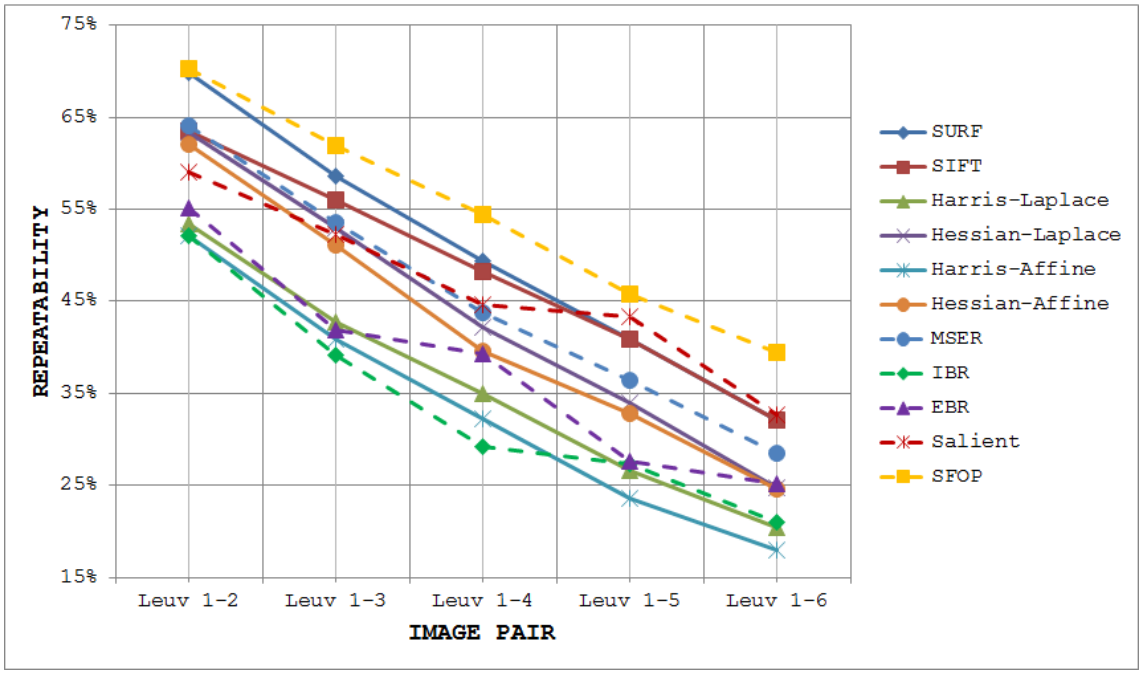

Figure 3-9: Repeatability results for state-of-the-art detectors for Leuven dataset (light) using proposed measure 1

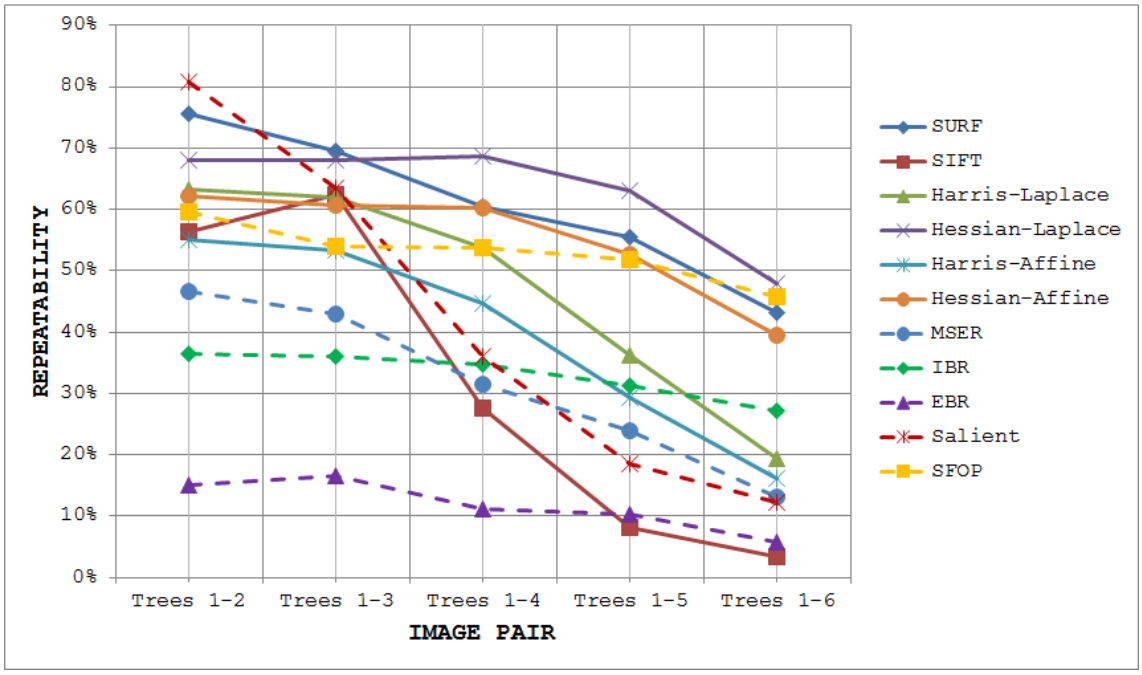

Figure 3-10: Repeatability results for state-of-the-art detectors for Trees dataset (blur) using proposed measure 1

UBC dataset. The effect of increasing JPEG compression on the repeatability scores of feature detectors is shown in Figure 3-11. These results seem to be largely consistent with those reported in [46]. Hessian-Laplace and Hessian-Affine perform best, followed by Harris-Laplace and Harris-Affine. SURF also shows relatively good performance. The repeatability scores achieved by SIFT decrease rather substantially with increasing JPEG compression.

Wall dataset. This dataset contains a textured scene with increasing viewpoint changes. Figure 3-12 depicts the results obtained for this dataset utilizing proposed measure 1. SIFT, SFOP and SURF out-perform other detectors for all image pairs in this dataset except the last one (Wall 1 and Wall 6). Interestingly, MSER, which is identified as the top performer in [46], achieves lower repeatability scores for most image pairs of this dataset when compared to Salient—the detector which is shown as the lowest ranked in [46]. Salient also out-performs EBR and IBR which again contradicts the previous findings [46].

General Conclusions. The author has evaluated eleven state-of-the-art detectors utilizing proposed measure 1. The results shown in Figure 3-5 to Figure 3-12 provide new insights into their behavior under different geometric and photometric transformations. It can be concluded that some of the observations support previous findings but most of them largely disagree with them. There is generally a continuous decline in detector performances with any increasing geometric or photometric transformation, which is what intuitively one would expect. It is quite noticeable that no detector achieves high repeatability scores for all the image transformations discussed above. MSER, which is identified as the best detector in [46], does not show promising results in this study except for the Graffiti dataset where it dominates all other detectors. It is also important to note that Salient, a detector which is generally ranked the lowest in [46], here either out-performs or has similar performance to MSER in most cases.

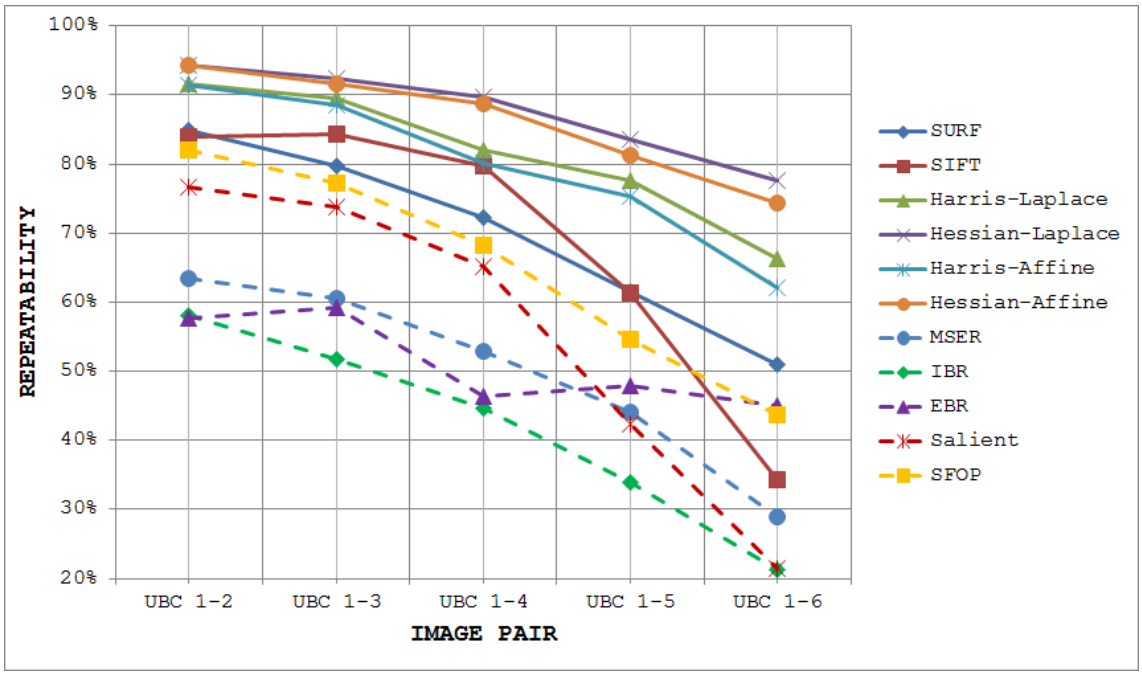

Figure 3-11: Repeatability results for state-of-the-art detectors for UBC dataset (JPEG compression) using proposed measure 1

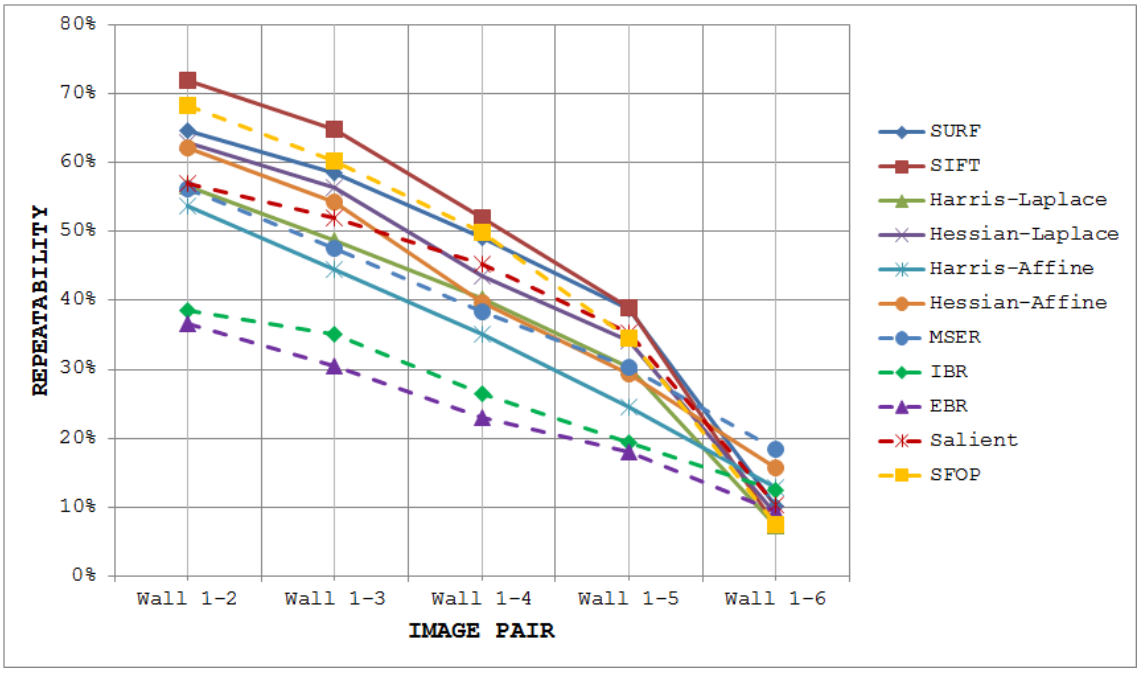

Figure 3-12: Repeatability results for state-of-the-art detectors for Wall dataset (viewpoint) using proposed measure 1

3.4.2 Results under Various Transformations using Proposed Measure 2

The comparative results for state-of-the-art detectors utilizing proposed measure 2 for the widely-used Oxford datasets [50] are shown in Figure 3-13–Figure 3-20. Although the repeatability scores in these results are generally higher than those presented in Section 3.4.1, the findings are largely consistent for the two evaluations done in this chapter, enabling us to draw the same general conclusions. SIFT seems to be the best detector for the Bark dataset in Figure 3-13. Hessian-Laplace and Hessian-Affine dominate for the Bikes dataset (see Figure 3-14). There is not much to distinguish between nearly all detectors for the Boat dataset in Figure 3-15. For the Graffiti dataset, MSER out-performs all other detectors (see Figure 3-16). SFOP and SURF achieve good repeatability scores for the Leuven dataset in Figure 3-17. SURF and Hessian-Laplace exhibit good performance for the Trees dataset (see Figure 3-18). For the UBC dataset in Figure 3-19, Hessian-Laplace and Hessian-Affine show high scores. Finally, SURF and SFOP perform well for the Wall dataset in Figure 3-20.

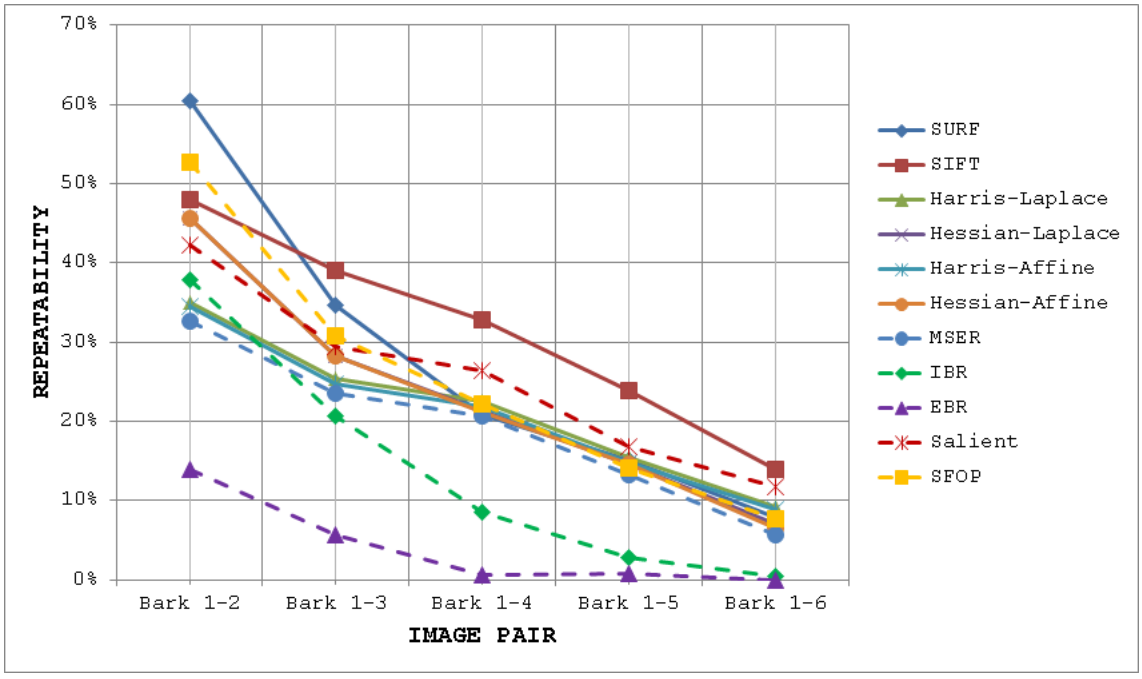

Figure 3-13: Repeatability results for state-of-the-art detectors for Bark dataset (zoom and rotation) using proposed measure 2

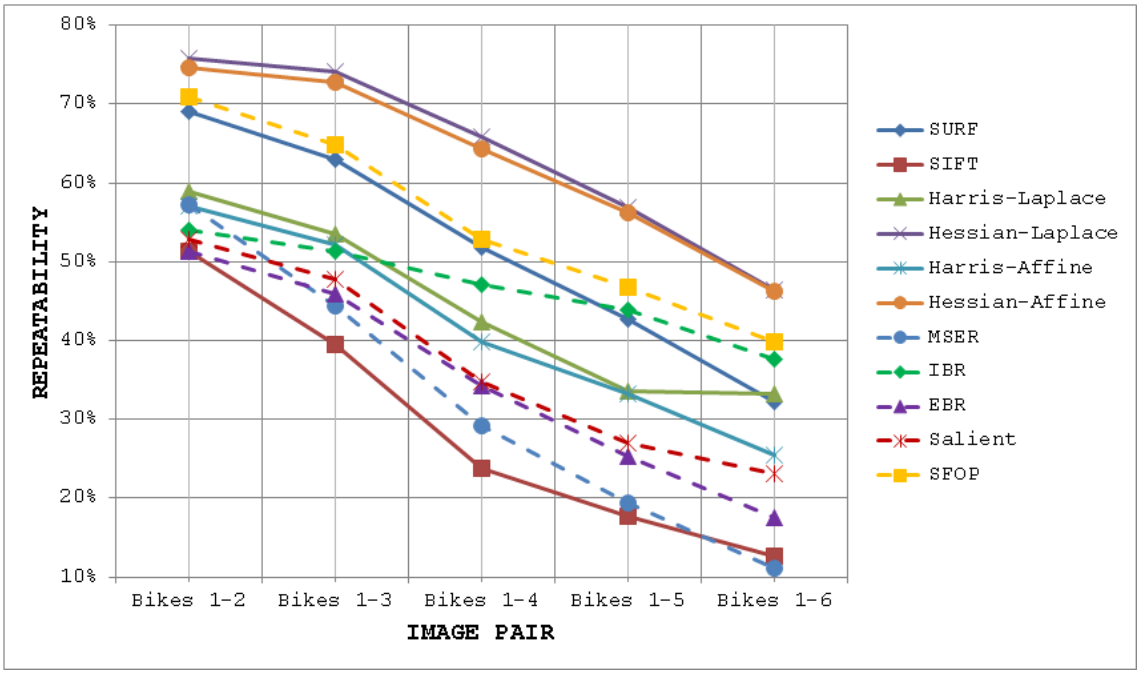

Figure 3-14: Repeatability results for state-of-the-art detectors for Bikes dataset (blur) using proposed measure 2

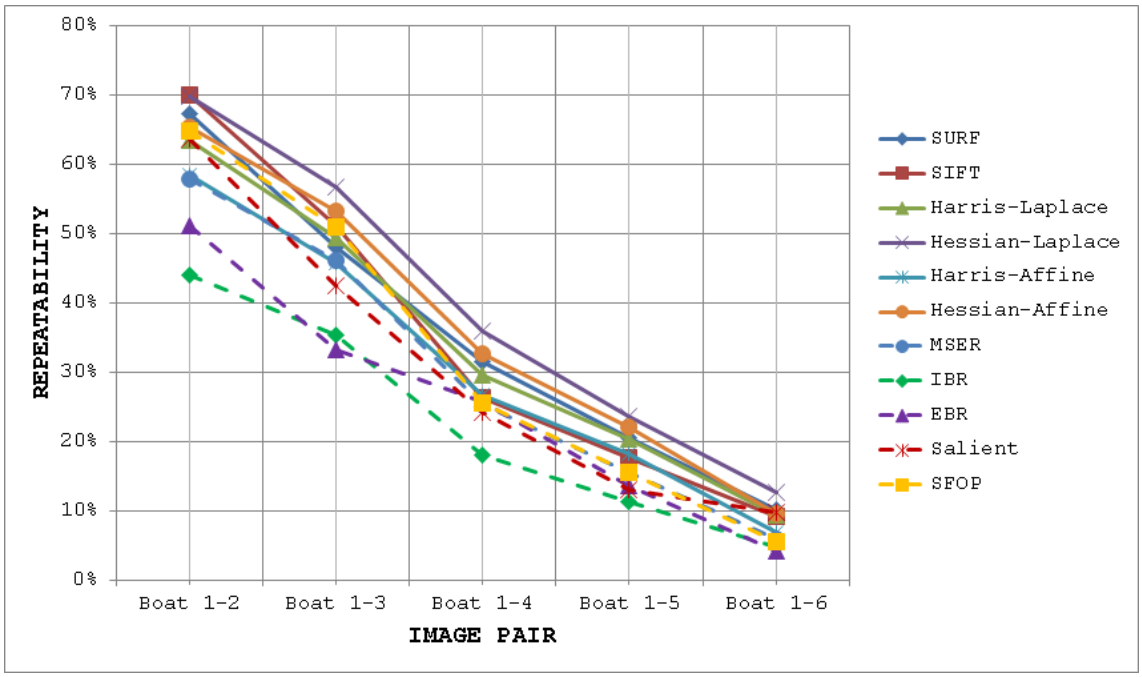

Figure 3-15: Repeatability results for state-of-the-art detectors for Boat dataset (zoom and rotation) using proposed measure 2

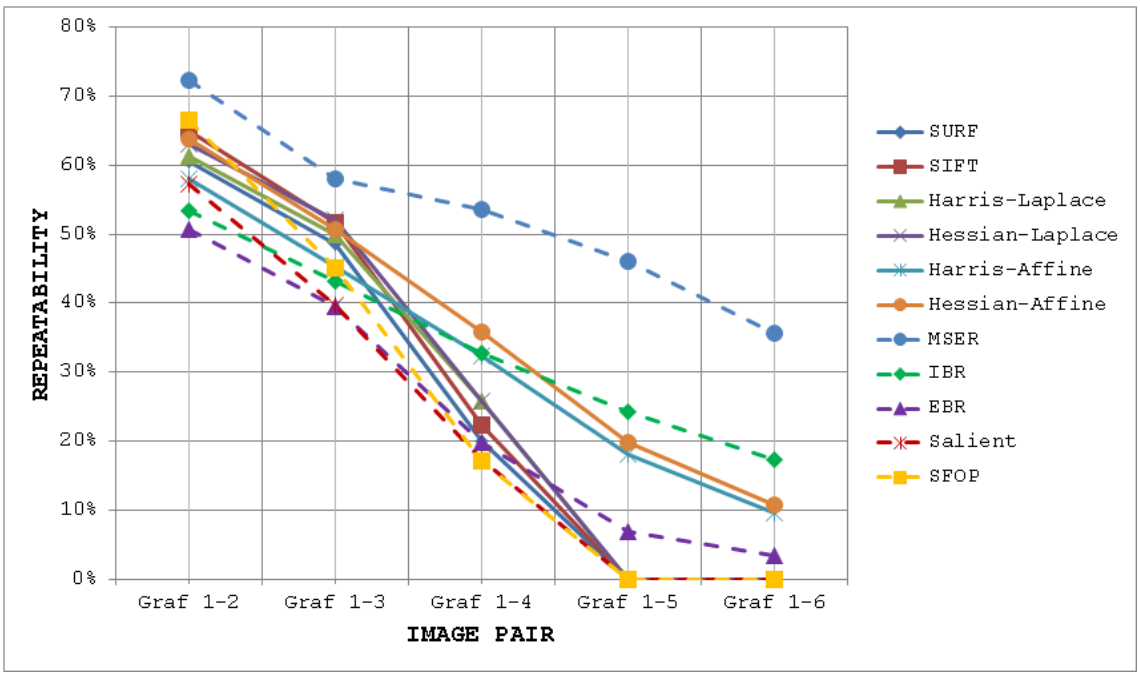

Figure 3-16: Repeatability results for state-of-the-art detectors for Graffiti dataset (viewpoint) using proposed measure 2

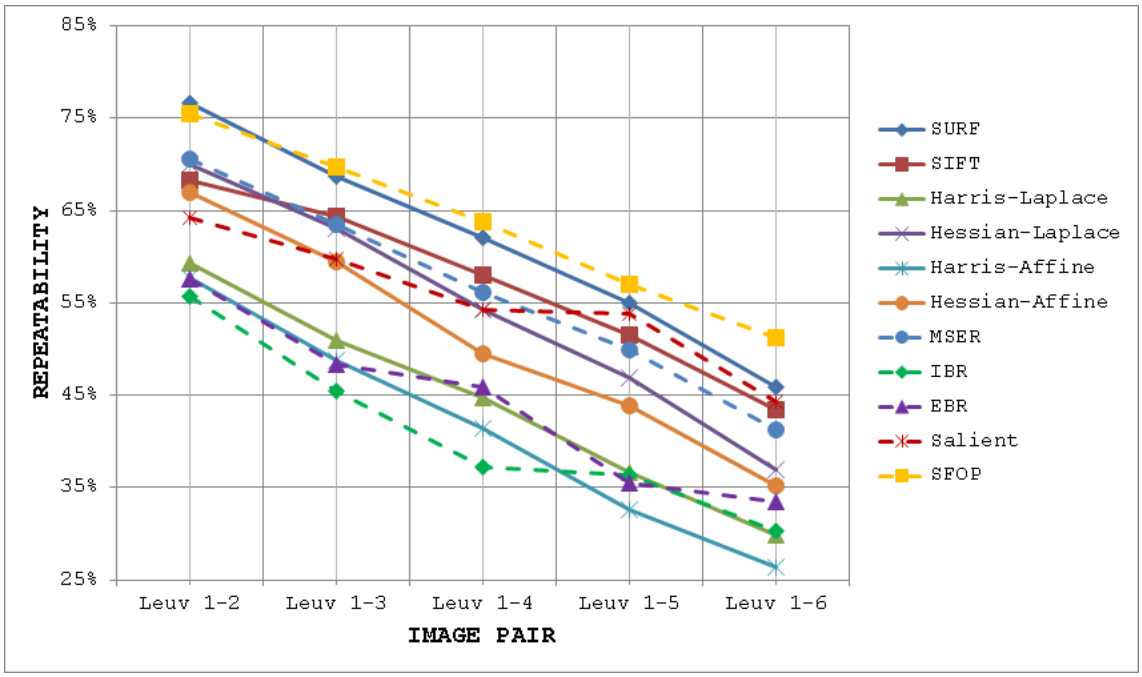

Figure 3-17: Repeatability results for state-of-the-art detectors for Leuven dataset (light) using proposed measure 2

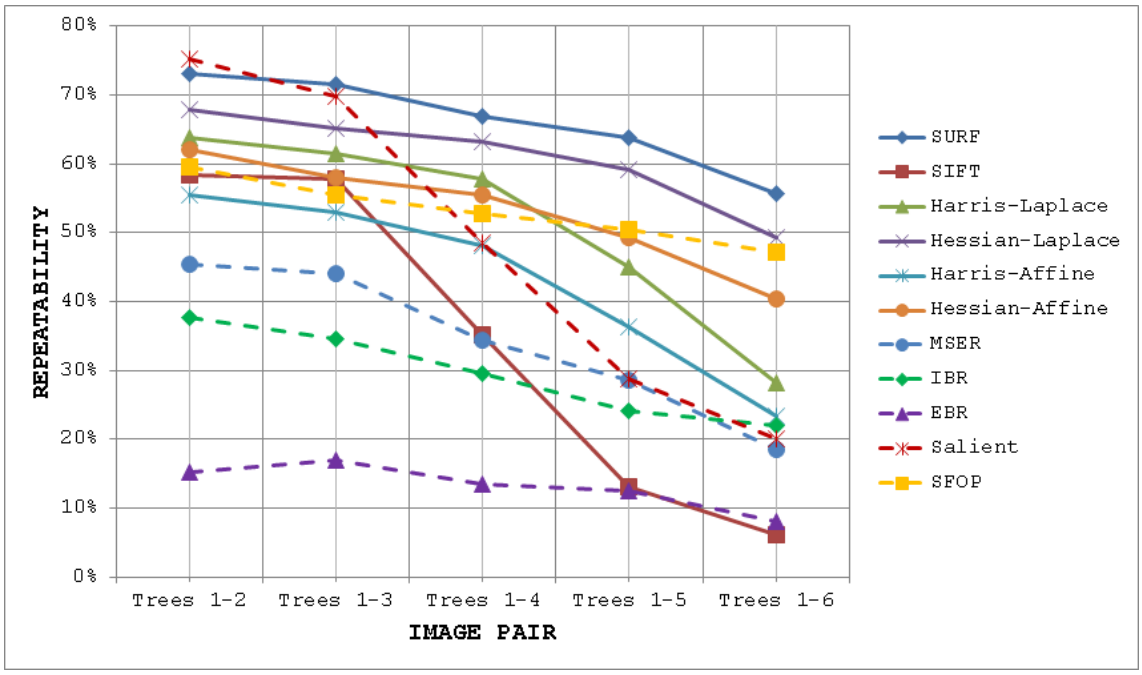

Figure 3-18: Repeatability results for state-of-the-art detectors for Trees dataset (blur) using proposed measure 2

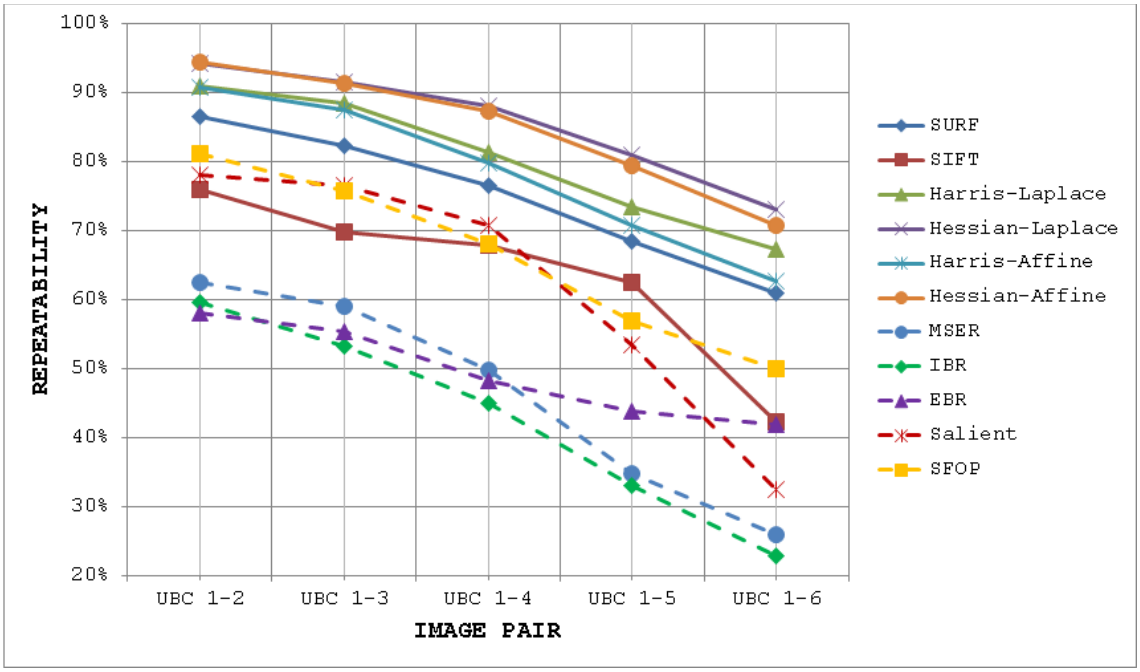

Figure 3-19: Repeatability results for state-of-the-art detectors for UBC dataset (JPEG compression) using proposed measure 2

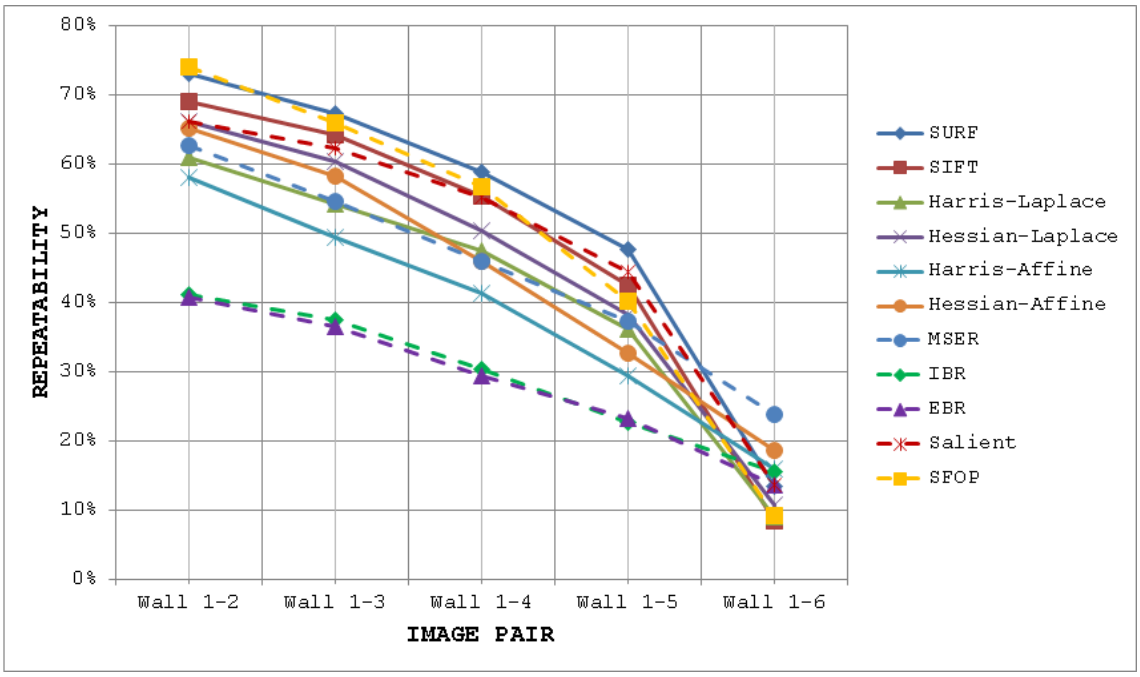

Figure 3-20: Repeatability results for state-of-the-art detectors for Wall dataset (viewpoint) using proposed measure 2

3.5 Summary

Repeatability, the most widely used performance metric for local feature detectors, does not always describe the true performance of the detector. This chapter has presented improved repeatability measures which provide results that *are* reliable and consistent with the actual performance of a wide variety of detectors across a number of well-established datasets. Based on the proposed measures, eleven state-of-the-art local invariant feature detectors were evaluated using standard datasets. The results obtained largely contradict the previous findings and provide new performance scores for these popular feature detectors under various image transformations. These performance curves are more consistent with what experienced vision researchers expect and encounter.

4 Repeatability: A Systems Design Perspective

The value of an idea lies in the using of it.

THOMAS EDISON

By utilizing one of the improved repeatability measures proposed in Chapter 3, a generic framework is presented in this chapter that allows assessment of the upper and lower bounds of detector performance and finds statistically significant performance differences between detectors as a function of image transformation amount by introducing a new variant of McNemar's test in an effort to design more reliable and effective vision systems. The proposed framework is then employed to establish *operating* and *guarantee* regions for several state-of-the art detectors and to identify their statistical performance differences for three specific image transformations: JPEG compression, uniform light changes and blurring. The results are obtained using newly acquired, large image databases for JPEG compression (7546 images with 539 different scenes), blur (5390 images with 539 different scenes) and uniform illumination changes (7546 images with 539 different scenes). For improving performance in the presence of uniform light variations, this chapter proposes including a pre-processing step as part of any feature detection technique. It demonstrates that this technique improves the performance of state-of-the-art detectors significantly in the presence of uniform light variations.

4.1 Introduction

Consider designing a small power supply for one of the coldest inhabited regions on the planet – Oymyakon, a village in Russia. Apart from some general design constraints, such as the required maximum output voltage, freezing temperatures in the area of deployment make this design task more challenging as the characteristics of most electronic components change with temperature. For example, the capacitance of a capacitor is a function of temperature. Similarly, the current-voltage characteristics of a diode are also dependent upon temperature. Thus, looking at the *datasheets* of the required electronic components for this power supply would be a logical step for finding devices that operate reliably in extremely low temperatures. Only those components would be selected which show stable operating characteristics across the required range of temperatures to ensure that the final output of the power supply would satisfy the initial design specifications.

What is most impressive about the above design example is the methodical manner in which electronic systems are designed in general through accurate knowledge of the upper and lower performance limits of the required electronic components. The operating characteristics of every device to be used in the designed system are well known through their datasheets which make it easier to predict the output of the system as a whole under different scenarios, such as large variations in temperature. The main motivating factor behind this approach is to make the designed system as much reliable as possible.

Now come back to the computer vision world and design a simple toy car tracking system with local feature detection as its primary stage while expecting only 20% uniform decrease in illumination. Looking at the repeatability results presented in [46] (which are widely considered the most comprehensive) for the Leuven dataset (which involves uniform changes in light), MSER appears to be the best option for achieving a reasonable value of repeatability (more than 60%) for this small

transformation amount. Now consider two sample images shown in Figure 4-1 which the designed vision system would encounter when deployed in the actual environment. The first image is the reference image and the second image has undergone 20% uniform decrease in light relative to the reference. Theoretically speaking, the feature detection unit (based on MSER) of the designed vision system would achieve high repeatability score for this negligible image transformation. As it turns out, MSER only manages a repeatability value of only 28.17% for the image pair shown, which is much less than what is expected of the feature detection unit and highlights its unreliable behavior – a stark contrast to the power supply design example.

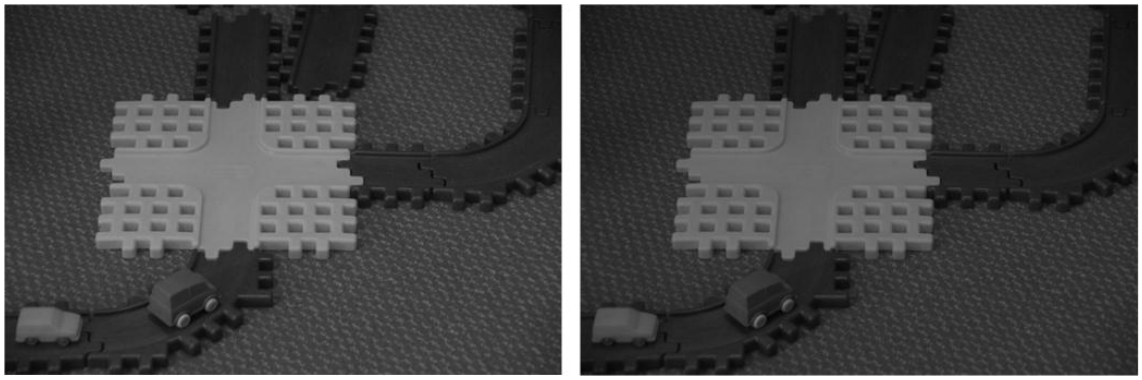

Figure 4-1: Two sample images; the left image is the reference image whereas the right image undergoes 20% uniform decrease in illumination

Having gone through the above two examples, the obvious question which springs to mind is: what is the distinguishing factor between the two approaches which makes one designed system highly reliable and the other one unpredictable? The answer is fairly simple. Every component of an electronic system has known operating characteristics (or performance limits). The system which is built using these components would continue to operate according to the design specifications. On the other hand, how much do we all know about the upper and lower performance bounds of the MSER detector, essentially the primary stage of the designed vision system, for 20% uniform decrease in illumination? Not more than what is reported in [46], which is based on a single dataset!

Indubitably, vision researchers can incorporate the good practices of electronic system designers and take a leaf out of their book. Building reliable and effective vision systems is the sole purpose of working in the area and perhaps this needs to be reiterated. Recently, it has been commented by [49] that the vision community places too much emphasis on beating latest benchmark numbers regardless of whether the improvement over other methods is statistically significant. This is especially true for local feature detection where most methods seem to have confined themselves to some particular datasets for demonstrating their superiority over other competing algorithms. This is essentially where the original objective of building a reliable vision system is lost. The author contends that it is time to shed the *best-case* analysis approach to concentrate on the original purpose.

For achieving the goal of reliability and effectiveness, the operating characteristics of every component in the vision system should be well known – something which is in line with the electronic system design practices *but* has not been done yet. The question is: how to determine the operating characteristics of different components in a vision system? Clearly, a principled framework, utilizing suitable metrics that mirror real-world performance of different components of the system, is required for this – something which is currently lacking.

This chapter attempts to bridge this research gap. Limiting itself to the feature detection stage, the chapter demonstrates how one of the improved repeatability measures proposed in Chapter 3 can be utilized from a vision systems design perspective. Inspired by the good practices of electronic systems design, this chapter proposes a generic framework for finding the *operating* and *guarantee* regions of a local feature detector under some specific image transformation. Taking into account the comments of [49], the framework also identifies statistically significant performance differences between detectors by introducing a variant of McNemar's test [51, 52]. To demonstrate the utility of this framework, three specific image transformations, namely JPEG compression, blurring, and uniform light

changes, are used. The *operating* and *guarantee* regions for several state-of-the art detectors are established and statistical performance differences are identified for these image transformations. These detailed results provide insights into the behavior of detectors, and are also useful from the vision systems design perspective. The chapter also presents large image databases for JPEG compression (7546 images with 539 different scenes), blur (5390 images with 539 different scenes) and uniform illumination changes (7546 images with 539 different scenes) which are utilized for obtaining the results for state-of-the-art feature detectors. Finally, the chapter proposes including a pre-processing step as part of any feature detection technique for improving its performance significantly under uniform changes in light.

The remainder of the chapter is structured as follows: Section 4.2 presents a generic framework utilizing one of the improved repeatability measures (see Section 3.3.3) for establishing the performance bounds of local feature detectors and for identifying statistically significant performance differences between them in an effort to build reliable vision systems. By employing the proposed framework and a newly acquired large image database, results for eleven state-of-the-art feature detectors under JPEG compression are presented in Section 4.2.2. To demonstrate the utility of the proposed framework, Section 4.4 and Section 4.5 present results for state-of-the-art feature detectors under blurring and uniform changes of illumination respectively. Section 4.6 proposes including a pre-processing step as part of any feature detection scheme to improve its performance in the presence of uniform light variations and backs it up by presenting results for several state-of-the-art detectors utilizing this method. A summary of the work described in this chapter is presented in Section 4.7.

4.2 Proposed Framework

Enthused by the good practices of electronic systems design for improving the reliability of the designed systems, this section presents a generic

framework targeting the feature detection stage of the vision system design process for developing systems that would follow the design specifications and would be more reliable and effective. Before discussing the details, it is worth stating that the proposed framework is based on two main principles:

1) The ability to determine the upper and lower performance bounds of a given detector under some specific type and amount of image transformation — an idea borrowed from electronic systems design practice.
2) The ability to identify statistically significant performance differences between a given detector and some other detector whose performance is considered a benchmark under specific type and amount of image transformation — a concept for taking into account the comment made by [49].

To adhere to the above-mentioned principles, the framework is divided into two distinct components. The details of these components are given below.

4.2.1 Component 1

The first part of the framework establishes the upper and lower performance bounds of a given detector. It utilizes one of the improved repeatability measures presented in Section 3.3.3 for achieving this objective. Assuming the availability of a large image database involving a specific type of image transformation with known ground truth mapping between images and consisting of n individual datasets with each having a different scene, the first component of the framework carries out the following steps:

1) The repeatability scores are computed using Equation 3-3 for all images in every individual dataset (of the large image database) by taking the first image in each dataset which contains no transformation, as the reference. Assuming that the amount of image transformation is varied in m discrete steps for every single dataset,

n values of repeatability are obtained for each discrete step. Let A be the set of m discrete steps representing specific transformation amounts

$$A = \{1, 2, 3, \ldots\ldots, m\} \qquad \textit{Equation 4-1}$$

Let B_k be the set of n repeatability values at any one specific step k, where k is an element of set A

$$B_k = \{b_{1k}, b_{2k}, b_{3k}, \ldots\ldots, b_{nk}\} \qquad \textit{Equation 4-2}$$

For example, if the image database consists of 539 different datasets (the number which will be used in the next few sections), each consisting of a sequence of 14 images, the values of n and m will be 539 and 14 respectively. In other words, there will be 539 values of repeatability available for each step of image transformation amount.

2) For every discrete step k, the maximum value of repeatability is

$$P = \{\max(B_1), \max(B_2), \max(B_3), \ldots\ldots, \max(B_m)\} \qquad \textit{Equation 4-3}$$

The values of set P are plotted against the corresponding image transformation amounts from set A to obtain a curve which represents the upper bound of performance for the given detector with variation in the amount of transformation. This curve is named the *max* curve.

3) For every discrete step k, the minimum value of repeatability is found to give

$$Q = \{\min(B_1), \min(B_2), \min(B_3), \ldots\ldots, \min(B_m)\} \qquad \textit{Equation 4-4}$$

The values of set Q are plotted against the corresponding image transformation amounts from set A to obtain a curve which represents the lower bound of performance for the given detector with the same variations in image transformation. This curve is named the *min* curve.

4) For every discrete step k, the median value of repeatability is found

$$S = \{\text{median}(B_1), \text{median}(B_2), \ldots., \text{median}(B_m)\} \quad \textit{Equation 4-5}$$

The values of set S are plotted against the corresponding image transformation amounts from set A to obtain a curve which represents the typical performance for the given detector with variation in image transformation amount. This curve is named the *median* curve.

5) By plotting the three curves together, the area between the *max* curve and the *min* curve is defined as the *operating region* of the detector. The detector is expected to produce repeatability scores that lie inside this region. A narrow *operating region* implies that the detector is stable and there is little variation between the maximum and minimum repeatability values that it can achieve for some specific amount of transformation. On the other hand, a large *operating region* indicates an unstable detector which may achieve high repeatability scores for some particular images but may fare poorly for others.

6) The area under the *min* curve is defined as the *guarantee region* of the detector. Repeatability values achieved by the detector should never be as low so as to lie inside this region. A wide *guarantee region* shows that the detector manages to achieve reasonably high repeatability values for every input image with increasing amount of image transformation. Contrary to that, a small *guarantee region* implies that the detector performs poorly with increasing amount of image transformation.

4.2.2 Component 2

The second part of the proposed framework identifies statistically significant performance differences between two given detectors by introducing a variant of the non-parametric McNemar's test [51, 52]. McNemar's test is a form of chi-squared test with one degree of freedom that

evaluates the performance of two algorithms based on their outcomes on a case-by-case basis over the same dataset [51, 52]:

$$Z = \frac{|N_{sf} - N_{fs}| - 1}{\sqrt{N_{sf} + N_{fs}}} \qquad \textit{Equation 4-6}$$

where N_{sf} and N_{fs} are the numbers of occurrences when one algorithm succeeds and the other algorithm fails. If $N_{sf} + N_{fs} \geq 30$, the statistic is reliable and Z can be converted into a probability using tables [51, 52].

For utilizing McNemar's test, a criterion is needed to determine whether a test case results in success or failure. This framework utilizes repeatability score as this criterion. By selecting a specific threshold value of the repeatability score, it is possible to determine the success or failure of a detector. However, there is a variety of feature detectors available which show large variations in absolute and relative performances for different types of image transformations, so it is difficult to select one specific repeatability threshold that would work for all cases without introducing any bias. To solve this problem, this framework introduces a variant of McNemar's test. Instead of fixing the threshold value for repeatability, this variant utilizes a *ROC-like* approach, where the value of the threshold is varied in t discrete steps for each specific image transformation amount. Let X be the set of discrete steps that represent specific test thresholds

$$X = \{1, 2, 3, \ldots., t\} \qquad \textit{Equation 4-7}$$

For any value k of set A, which represents the image transformation amount, there will be t *Z-scores* obtained for the two given detectors. This may be represented mathematically as:

$$Y_k = \{y_{1k}, y_{2k}, y_{3k}, \ldots\ldots., y_{tk}\} \qquad \textit{Equation 4-8}$$

By varying the value of k in the above equation within the range of set A allows *Z-scores* obtained for the two detectors to be viewed as a function of image transformation amount and test threshold. This can conveniently be displayed in the form of an image.

4.3 Results for JPEG Compression

To demonstrate the utility of the proposed generic framework, this section presents results for eleven state-of-the-art feature detectors which establish their *operating* and *guarantee* regions and identify statistically significant performance differences between them under JPEG compression.

4.3.1 JPEG Image Database

In [46], the authors examined the performance of different local feature detectors on the basis of a single dataset (UBC [50]) for JPEG compression ratios varying from 60% to 98%. To investigate the behavior of local feature detectors by employing the framework proposed in the previous section reliably, a much larger database of images with variation in JPEG compression ratio is required. Since there is no such resource available, this section presents a newly acquired database of images with JPEG compression ratios varying from 0% to 98%. The database consists of 7546 images with 539 different planar scenes, captured by the author; both structured and textured scenes are included to eliminate any potential dataset bias. It should be noted that the maximum number of scenes which has been used so far for studying the performance of local feature detectors under different image transformations is only 60 [177]; the number of scenes for the presented database is thus nearly 9 times that of what is used in [177]. Moreover, the scenes employed in [177] are not real-world scenes and are captured in a highly controlled environment, whereas the presented database consists of images with scenes that are encountered routinely in everyday life. Some images from the JPEG image database are shown in Figure 4-2. For every scene, the JPEG compression ratio is varied in 14 discrete steps from 0% to 98% (14 x 539 = 7546). The database was generated using the *cjpeg* and *djpeg* utilities in a Linux-based environment by varying the image quality parameter. Each image in the database consists of 717 x 1080 pixels. Since there is no geometric transformation involved in the case of JPEG compression, the ground truth homography

relating any two images of the same scene with different compression ratios is simply a 3 x 3 identity matrix. To facilitate future research in this area, the image database is made available at [181].

Figure 4-2: Some images from the JPEG image database

4.3.2 Establishing Operating and Guarantee Regions

Results for eleven state-of-the-art detectors utilizing the framework proposed in Section 4.2 are presented in Figure 4-3 to Figure 4-13. These results determine the upper and lower bounds of performance of detectors with varying JPEG compression ratio, and then establish their *operating* and *guarantee* regions. This approach of presenting performance limits is intended to provide information for the design of robust vision systems; it is entirely possible that the standard deviations of performance may be significantly smaller than these limits. Before discussing the results, it is worth stating that this appears to be the first attempt to do such a detailed analysis; there is no other work in the literature with which these findings can be compared to determine consistencies and contradictions. The results

provide useful insight into the behavior of detectors under JPEG compression.

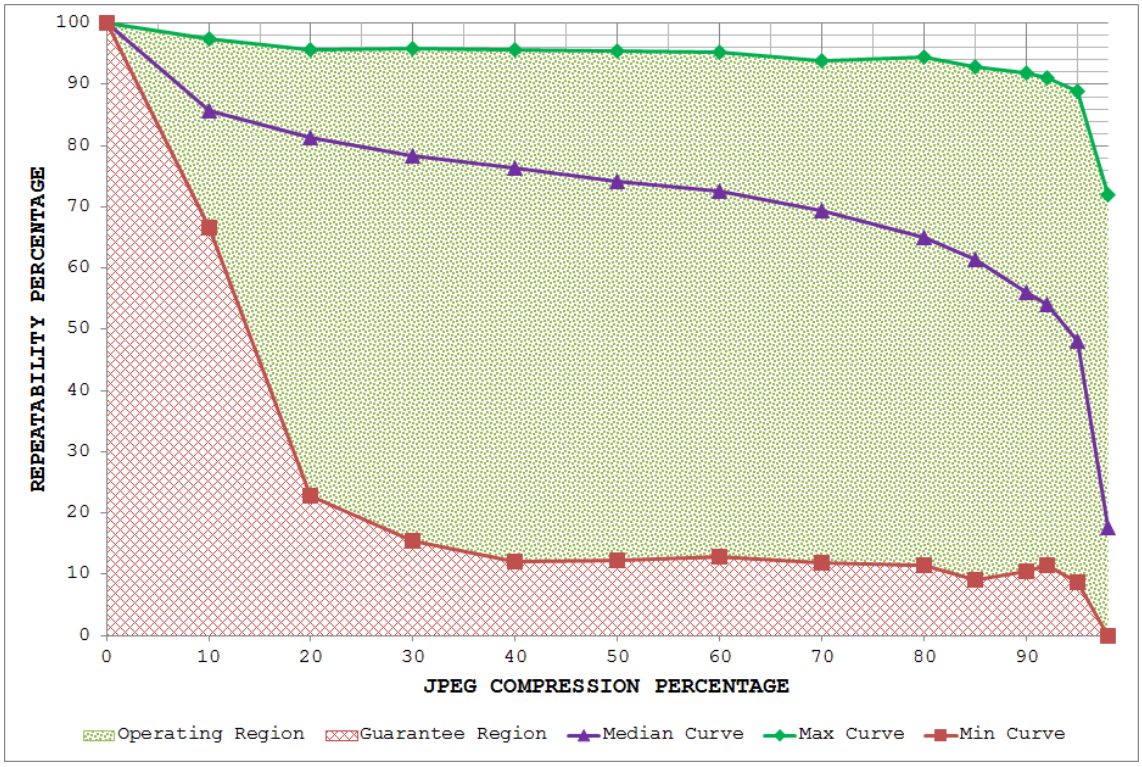

Figure 4-3: JPEG database results for MSER utilizing the proposed framework

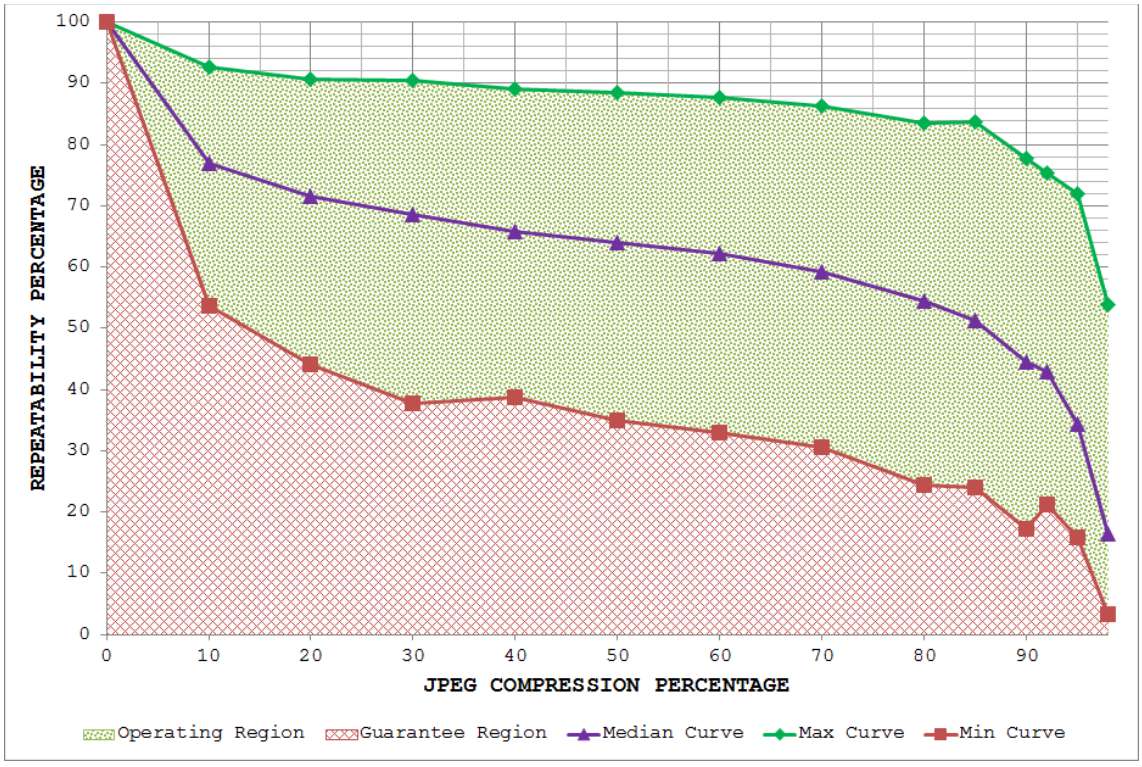

Figure 4-4: JPEG database results for IBR utilizing the proposed framework

Figure 4-3 depicts the *operating* and *guarantee* regions for the MSER detector. Although MSER does perform well for some particular images with increasing JPEG compression ratio (see the max curve in Figure 4-3), it is evident from the large *operating region* that its behavior is unstable. Even for small amounts of transformation, MSER fails to achieve high repeatability values for some images (see the value of min curve at 20% JPEG compression). Such an unpredictable performance does not make MSER a suitable choice for vision systems with more than 10% JPEG compression ratios as the detector may perform poorly for some images encountered.

The results for IBR are presented in Figure 4-4. IBR is more stable than MSER with increasing JPEG compression ratios as its *operating region* is smaller. It is however noticeable that IBR fails to beat the max curve of MSER. Although reasonable values of repeatability are achieved by IBR for JPEG compression ratios up to 95%, the performance may go to nearly zero for some particular images for 98% compression ratio.

Figure 4-5 shows the results for Salient utilizing the proposed generic framework. A much wider *guarantee region* for JPEG compression ratios up to 20% indicates that Salient is relatively more stable compared to MSER and IBR up to this point. For ratios greater than that, the *operating region* for Salient becomes wider and the min curve nearly goes to zero for 98% JPEG compression ratio. From a vision systems design perspective, Salient is a not an appropriate option if the JPEG compression ratio is expected to be more than 20%.

It is evident from Figure 4-6 that EBR shows highly unstable behavior. The *operating region* is wide, indicating that the performance of EBR may vary between rather high and rather low repeatability values for increasing JPEG compression ratios, depending upon the image content. Again, such behavior is not desirable when designing vision systems as it jeopardizes the final output of the system. Thus, EBR does not appear to be the best detector, even for vision systems expecting small JPEG compression ratios.

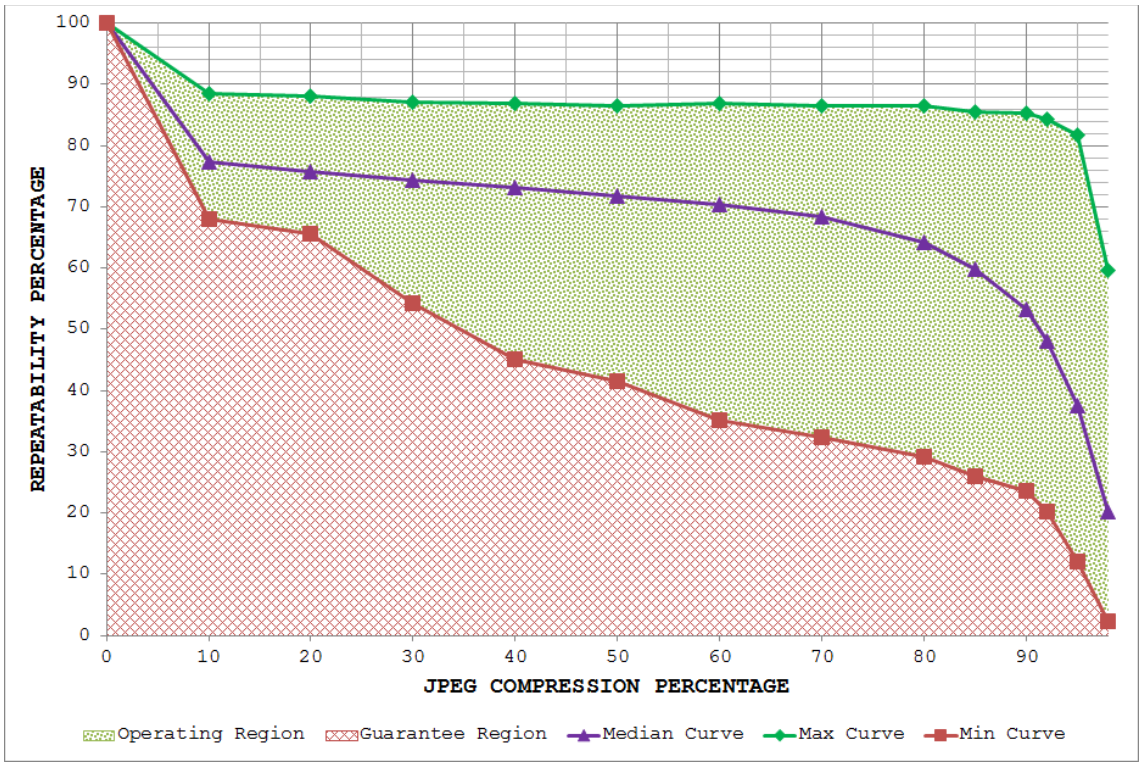

Figure 4-5: JPEG database results for Salient detector utilizing the proposed framework

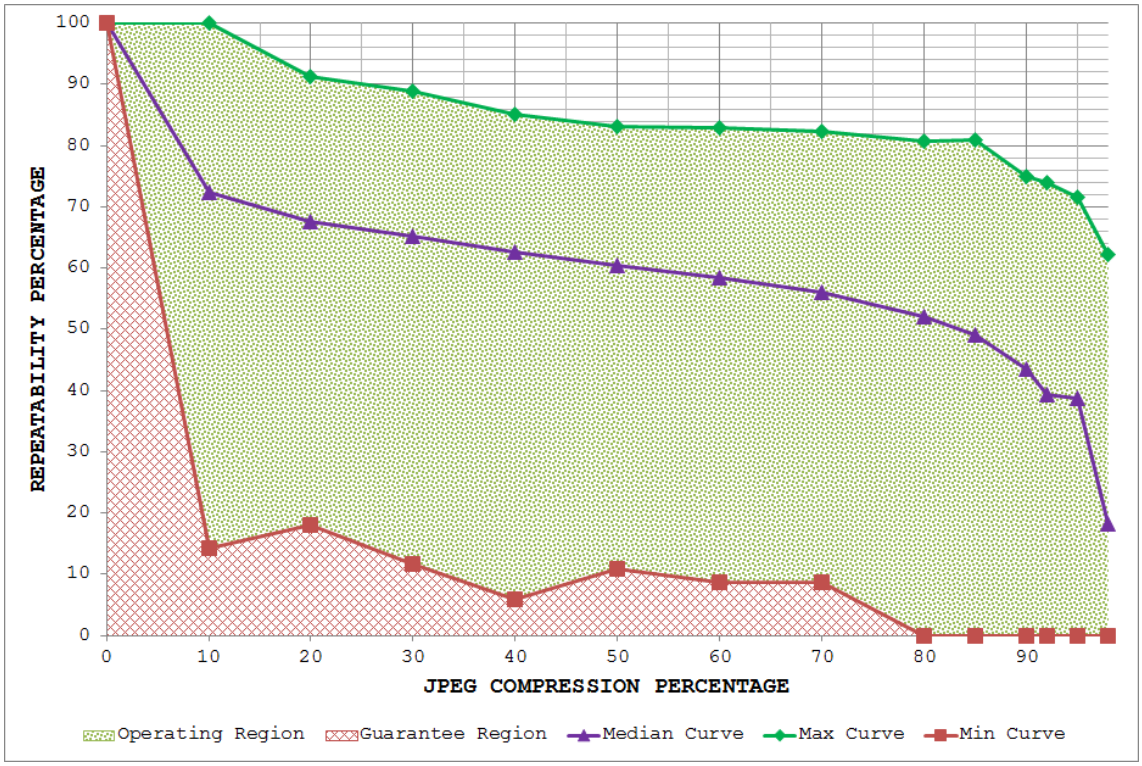

Figure 4-6: JPEG database results for EBR utilizing the proposed framework

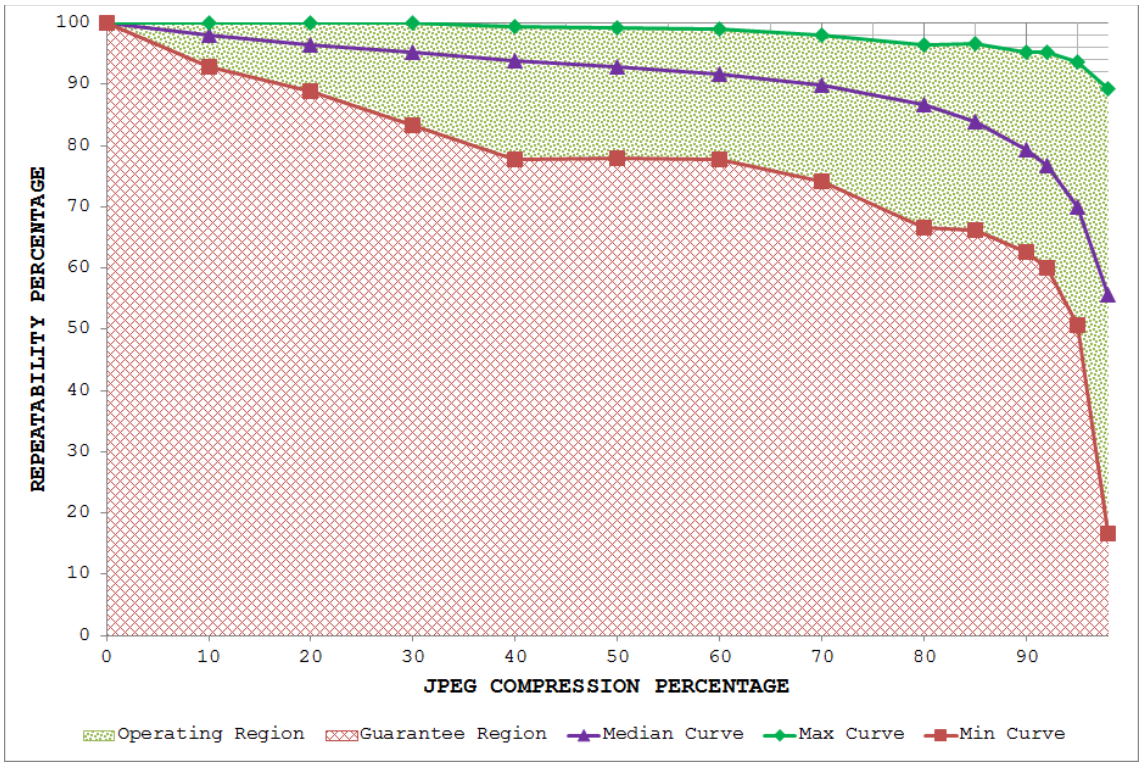

Figure 4-7: JPEG database results for SURF detector utilizing the proposed framework

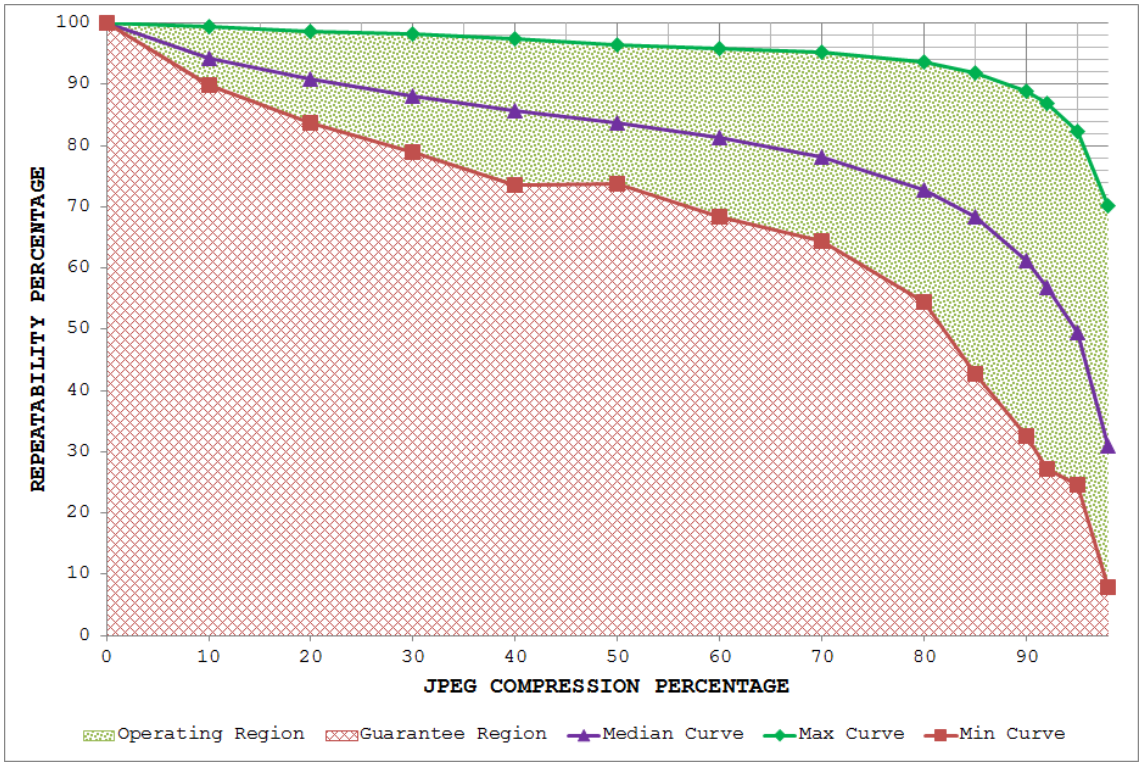

Figure 4-8: JPEG database results for SFOP utilizing the proposed framework

The *operating* and *guarantee* regions for SURF are shown in Figure 4-7. SURF performs well for increasing JPEG compression ratios up to 95% as is evident from its wide *guarantee region*. It shows relatively poor stability only for the case when JPEG compression ratio is 98%. This makes SURF a good choice from a vision systems design perspective when the expected JPEG compression ration does not exceed 95%.

Figure 4-8 depicts the results for SFOP. It is clear that the detector experiences small but continuous degradation in performance with increasing JPEG compression ratios; indeed, its narrow *operating region* for compression ratios up to 60% indicates that the detector is quite stable up to this point. For JPEG compression ratios exceeding 90%, SFOP may fail to achieve repeatability scores greater than 35%, depending upon the image content.

The results for Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine utilizing the proposed framework are presented in Figure 4-9 to Figure 4-12 respectively. The narrow *operating regions* for these detectors, especially Hessian-Laplace and Hessian-Affine, demonstrate their stability to increasing JPEG compression. These four detectors also have wide *guarantee regions,* which indicate that they manage to achieve high repeatability scores even for large JPEG compression ratios. Although the performance of these detectors fall sharply when the compression ratio exceeds 85%, the repeatability scores are still reasonable compared to all other detectors but SURF.

Finally, the operating and guarantee regions for SIFT are depicted in Figure 4-13. Although the performance of SIFT is reasonable, its operating region is wider than those of Hessian-Laplace and Hessian-Affine and grows with increasing JPEG compression ratio. Moreover, the performance of SIFT may go to nearly zero depending upon the image content for 98% compression ratio (see the min curve in Figure 4-13). It may be concluded that SIFT is a suitable option for vision systems expecting JPEG compression ratios up to 50%.

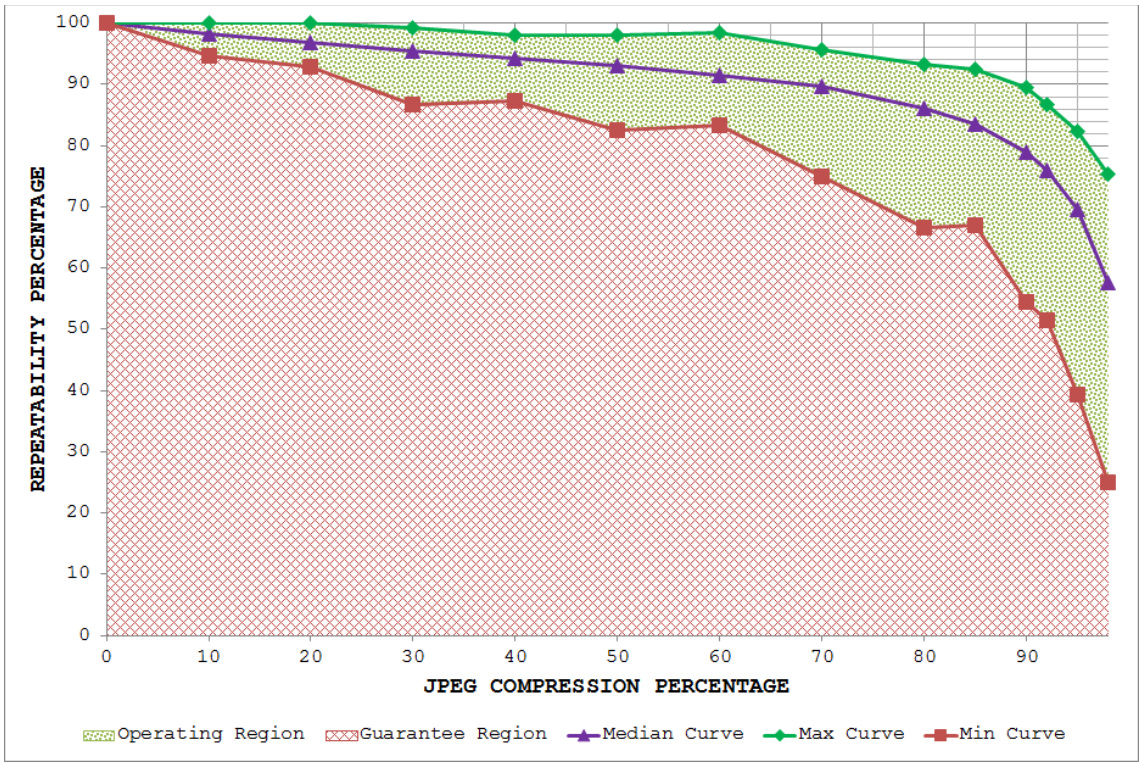

Figure 4-9: JPEG database results for Harris-Laplace utilizing the proposed framework

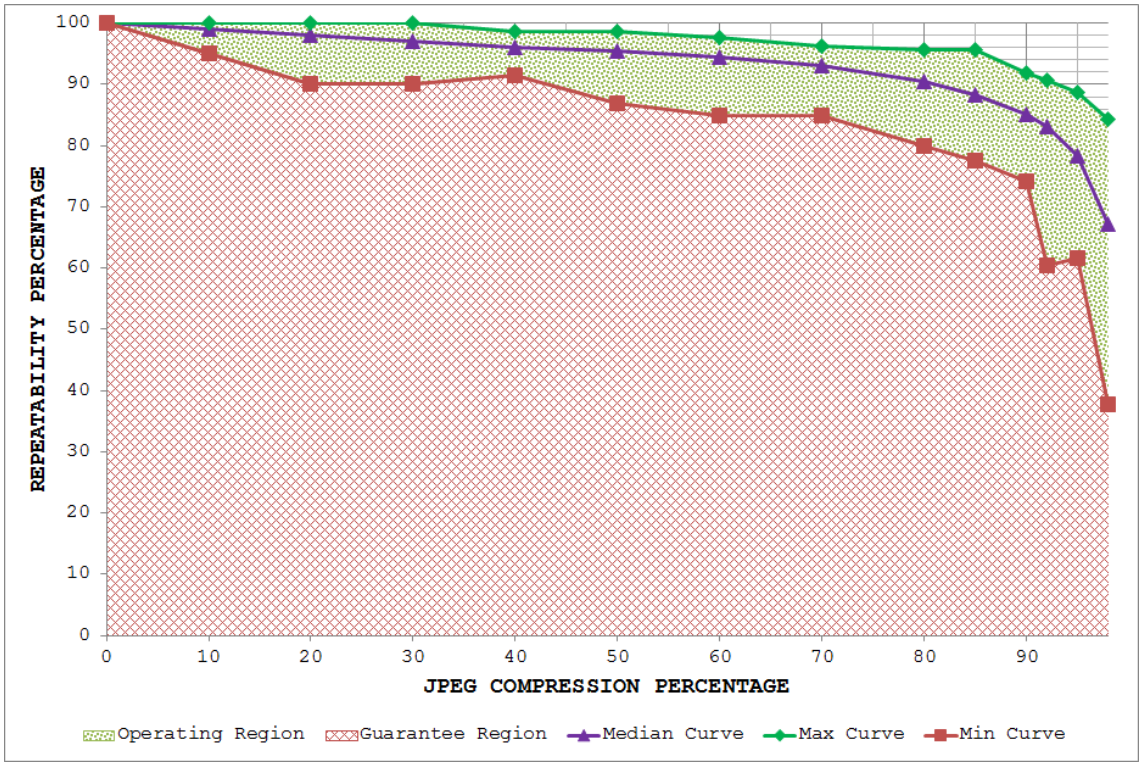

Figure 4-10: JPEG database results for Hessian-Laplace utilizing the proposed framework

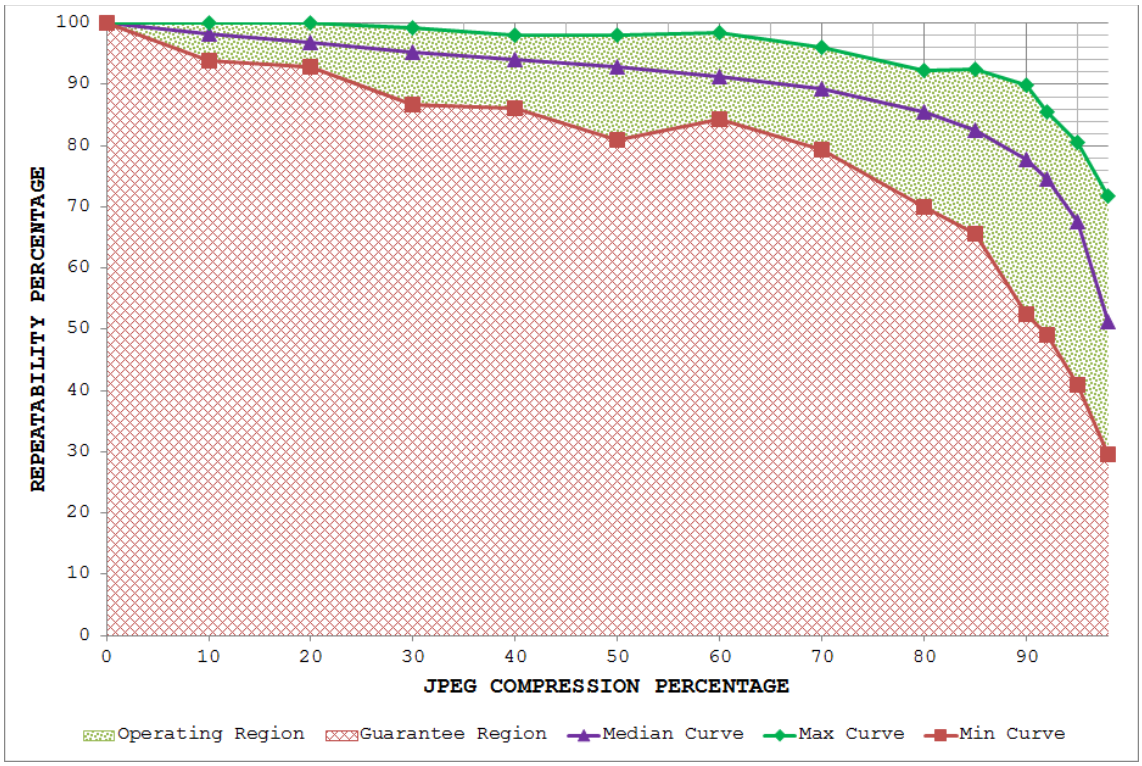

Figure 4-11: JPEG database results for Harris-Affine utilizing the proposed framework

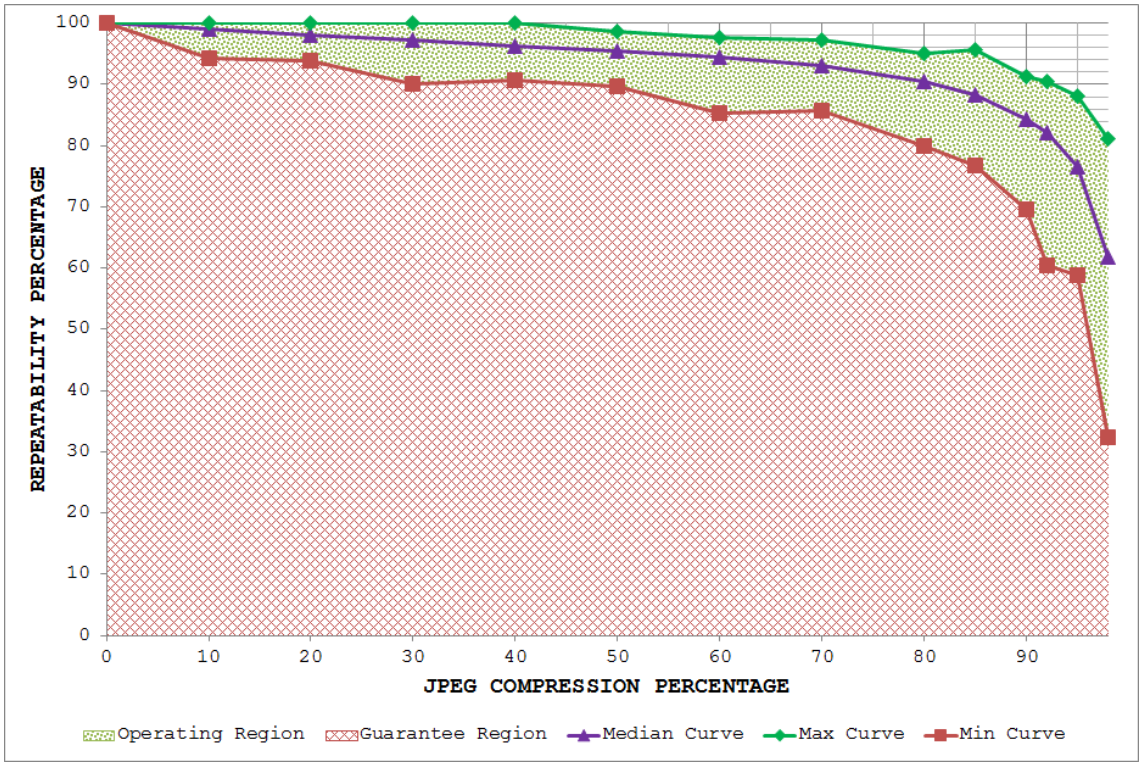

Figure 4-12: JPEG database results for Hessian-Affine utilizing the proposed framework

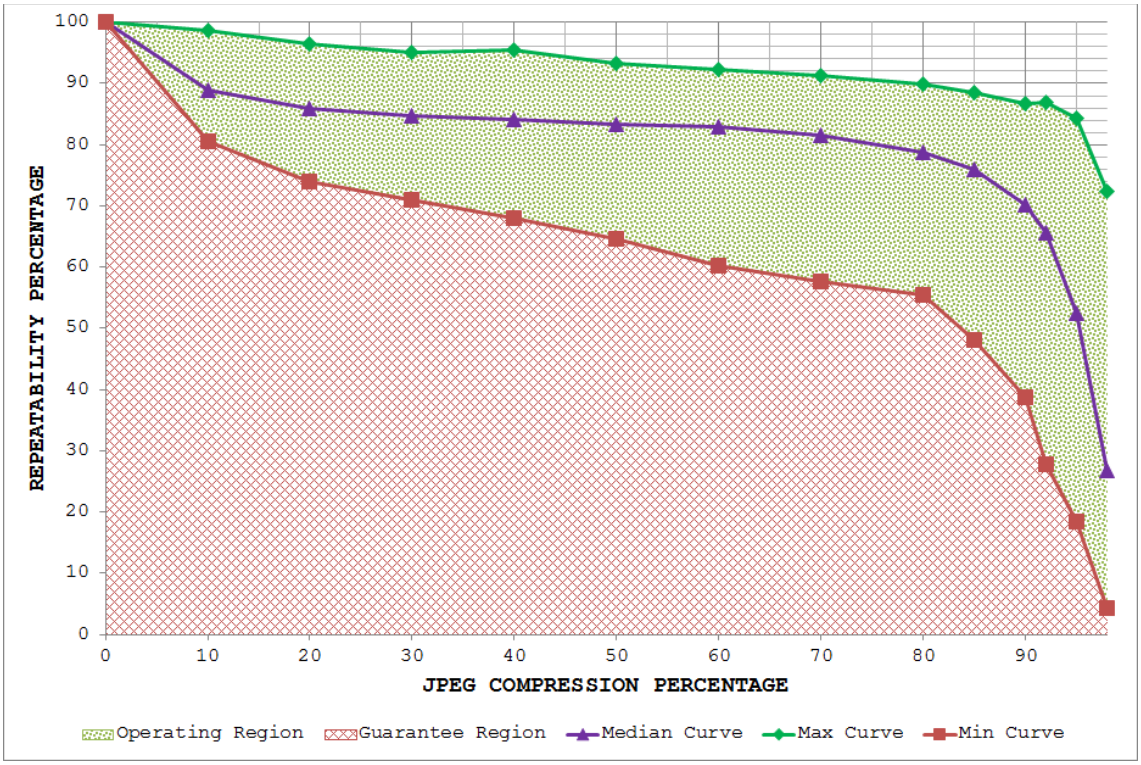

Figure 4-13: JPEG database results for SIFT detector utilizing the proposed framework

Not only do the results shown in Figure 4-3 to Figure 4-13 for JPEG compression provide insights into the behavior of detectors for this particular transformation, they also allow a vision system designer to select the best detector considering the design constraints for achieving more reliability — a feature inherited from electronic systems design practices.

4.3.3 Identifying Statistically Significant Performance Differences

For the statistical performance comparison of the feature detectors utilizing the proposed framework, results are presented in Figure 4-14 to Figure 4-17. Color coding in these figures indicate the *Z-scores* obtained as a function of image transformation amount and McNemar's test threshold when one detector is compared with another. Although the value of Z is always positive, here a sign convention has been used to distinguish the detector with the better performance of the two examined: a positive *Z-score* shows that the first detector is better than the second, whereas a negative value of Z indicates the converse.

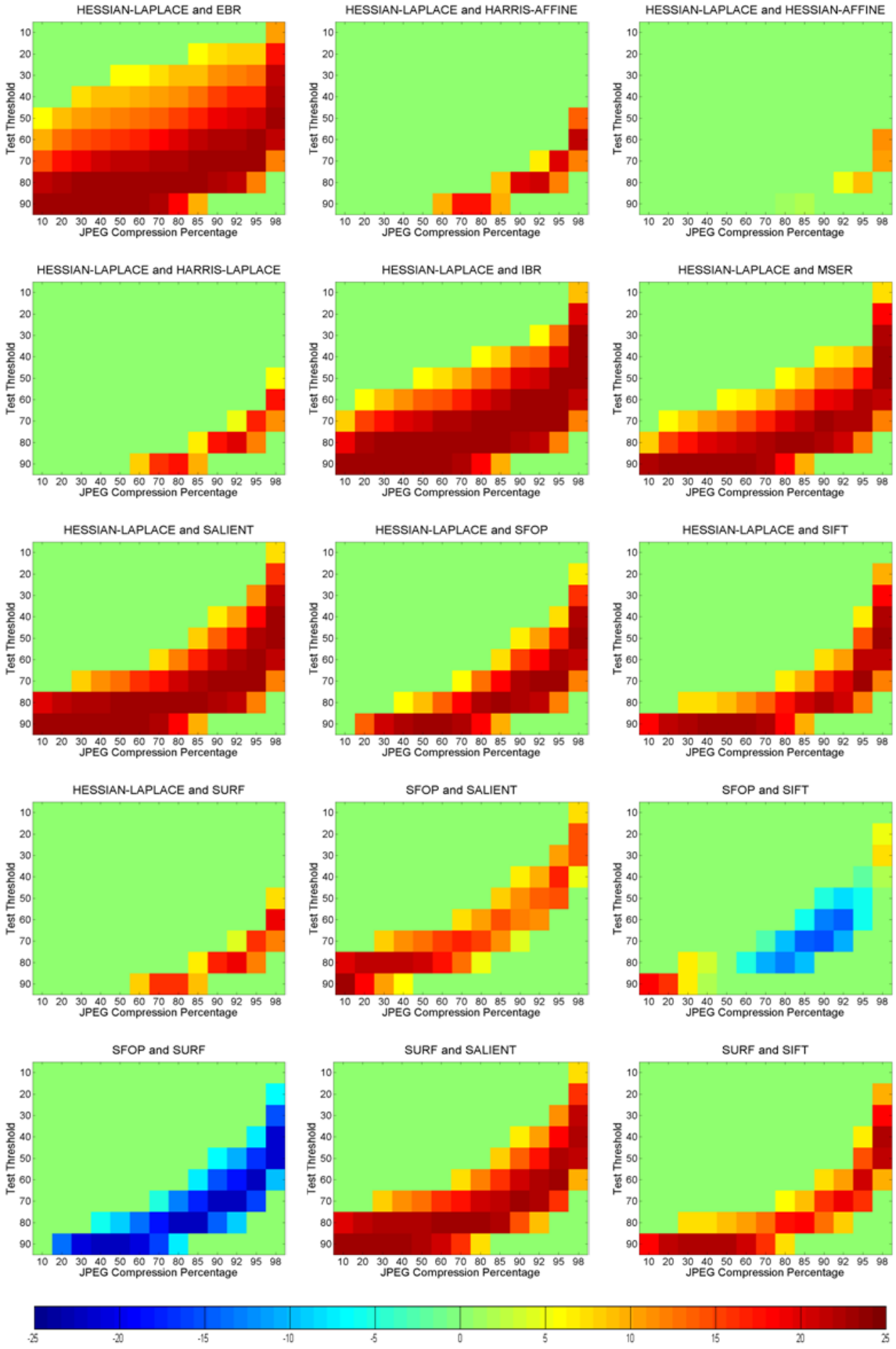

Figure 4-14: JPEG database results for Hessian-Laplace, SFOP and SURF with other detectors showing Z-scores obtained using the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas negative values show the converse

It is evident from Figure 4-14 that Hessian-Laplace out-performs EBR, IBR, MSER and Salient for most JPEG compression ratios when the test threshold is varied from 10% to 90% as indicated by large positive values of Z — a confirmation that the performance differences between Hessian-Laplace and these detectors are statistically significant. A comparison of Hessian-Laplace with Harris-Laplace, Harris-Affine, Hessian-Affine and SURF shows that they have broadly similar performance, although Hessian-Laplace does dominate them for some particular JPEG compression ratios at specific test thresholds. From Figure 4-14, it can be concluded that Hessian-Laplace also performs better than SFOP and SIFT.

The statistical comparison of SFOP and SIFT is interesting: the two detectors have largely similar performance but for some particular test thresholds and JPEG compression ratios, SFOP out-performs SIFT and vice versa. SURF is dominant when compared to SIFT, Salient and SFOP. It appears that EBR fails to achieve better performance than all other state-of-the-art feature detectors in most cases, as is evident from the large negative values of Z in Figure 4-15. IBR is also comprehensively outperformed by SURF, and to some extent by SFOP and SIFT.

Harris-Laplace and Harris-Affine show better performance than IBR, MSER, Salient, SIFT and SFOP for most test thresholds and JPEG compression ratios in Figure 4-16. Moreover, their performances are largely similar to each other and to those of SURF and Hessian-Affine. In Figure 4-17, Hessian-Affine out-performs MSER, IBR, Salient and SFOP. The large positive values of Z for some particular test thresholds and JPEG compression ratios indicate that the performance differences between Hessian-Laplace and SIFT are statistically significant, with the former appearing better of the two compared. SURF, SFOP and SIFT also seem to out-perform MSER. Finally, for some particular test thresholds and JPEG compression ratios in Figure 4-17, Salient performs better than MSER and vice versa.

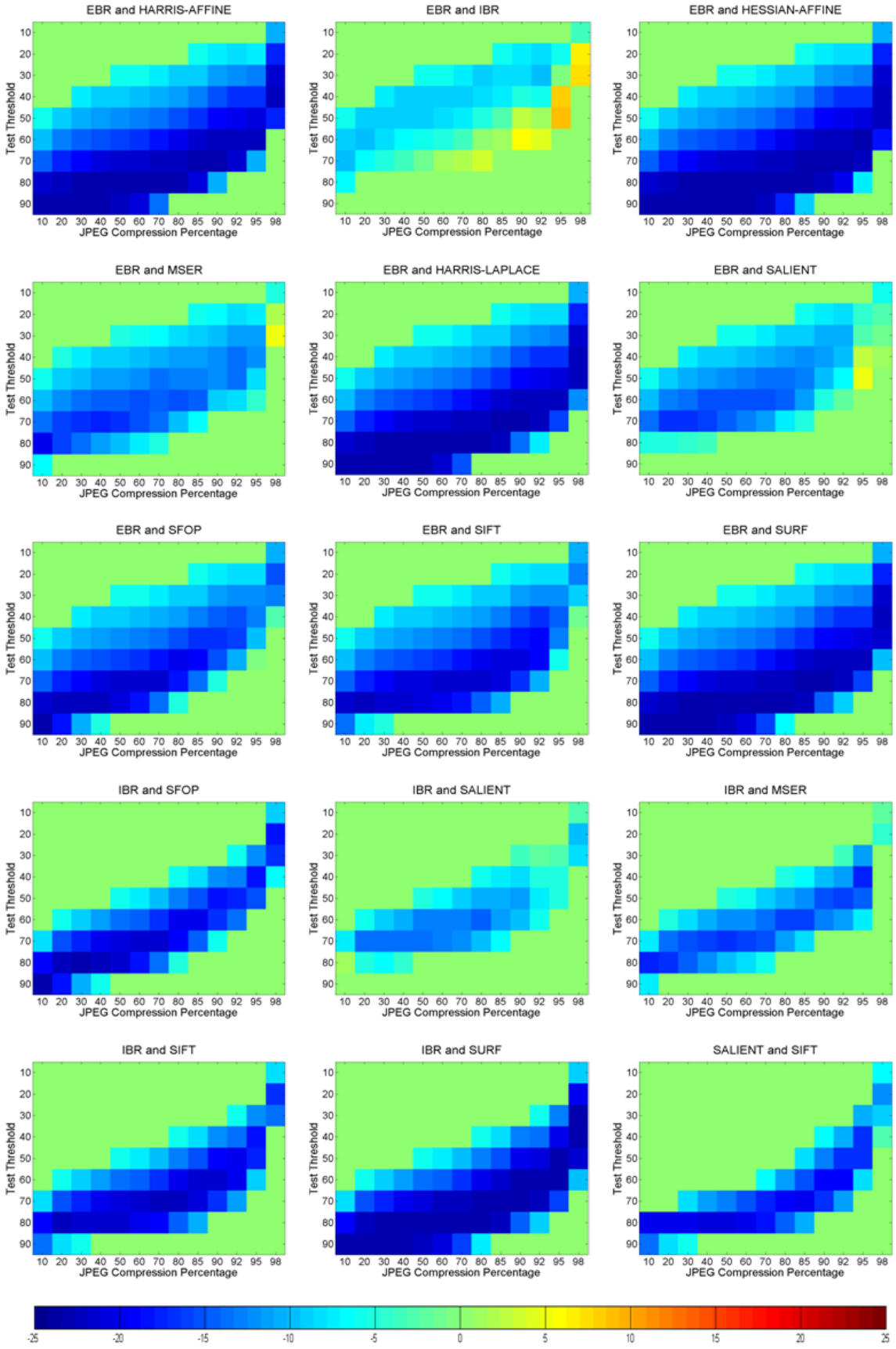

Figure 4-15: JPEG database results for EBR, IBR and Salient with the other detectors showing Z-scores obtained utilizing the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas the negative values show the converse

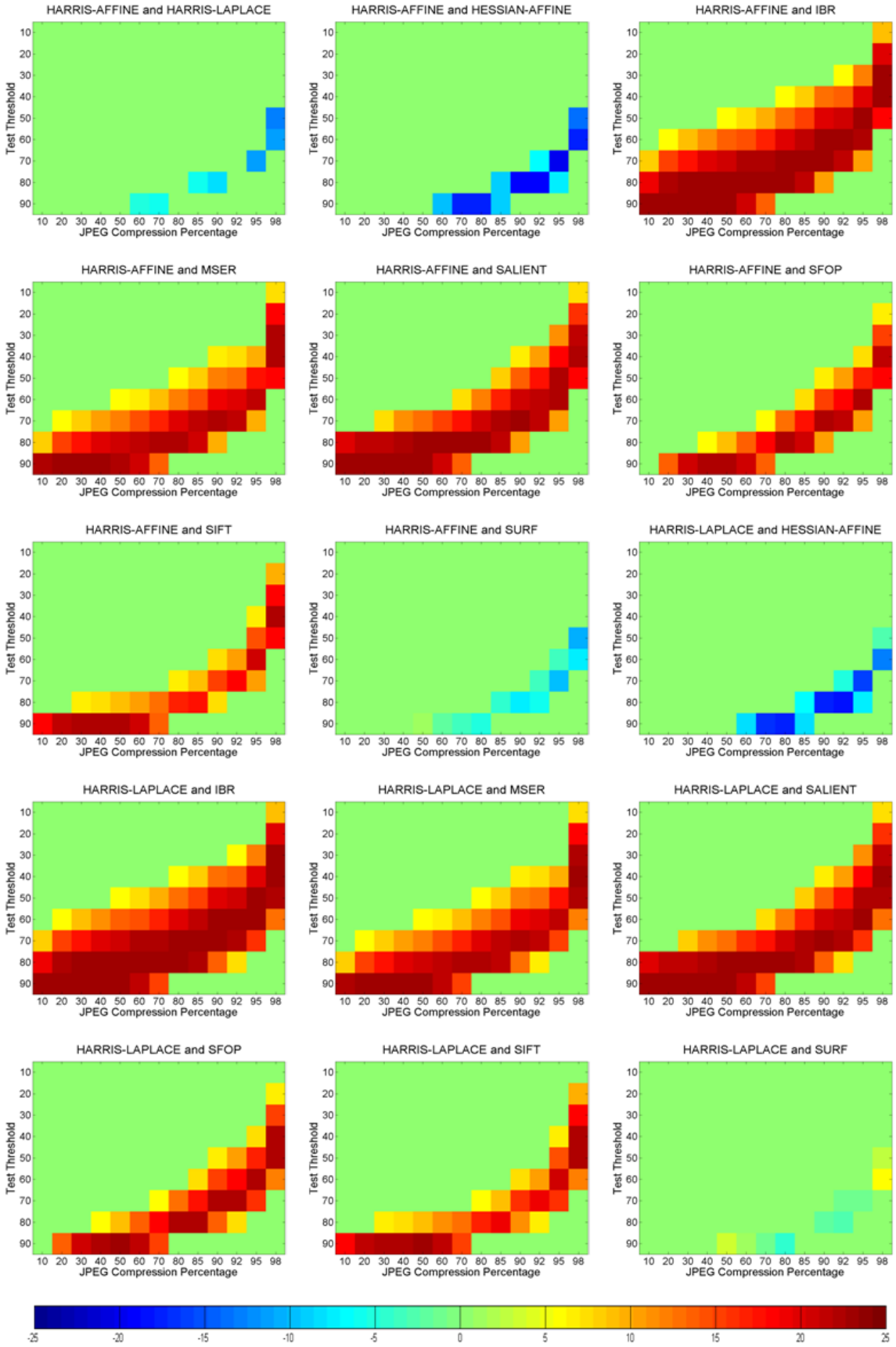

Figure 4-16: JPEG database results for Harris-Laplace and Harris-Affine with other detectors showing Z-scores obtained using the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas negative values show the converse

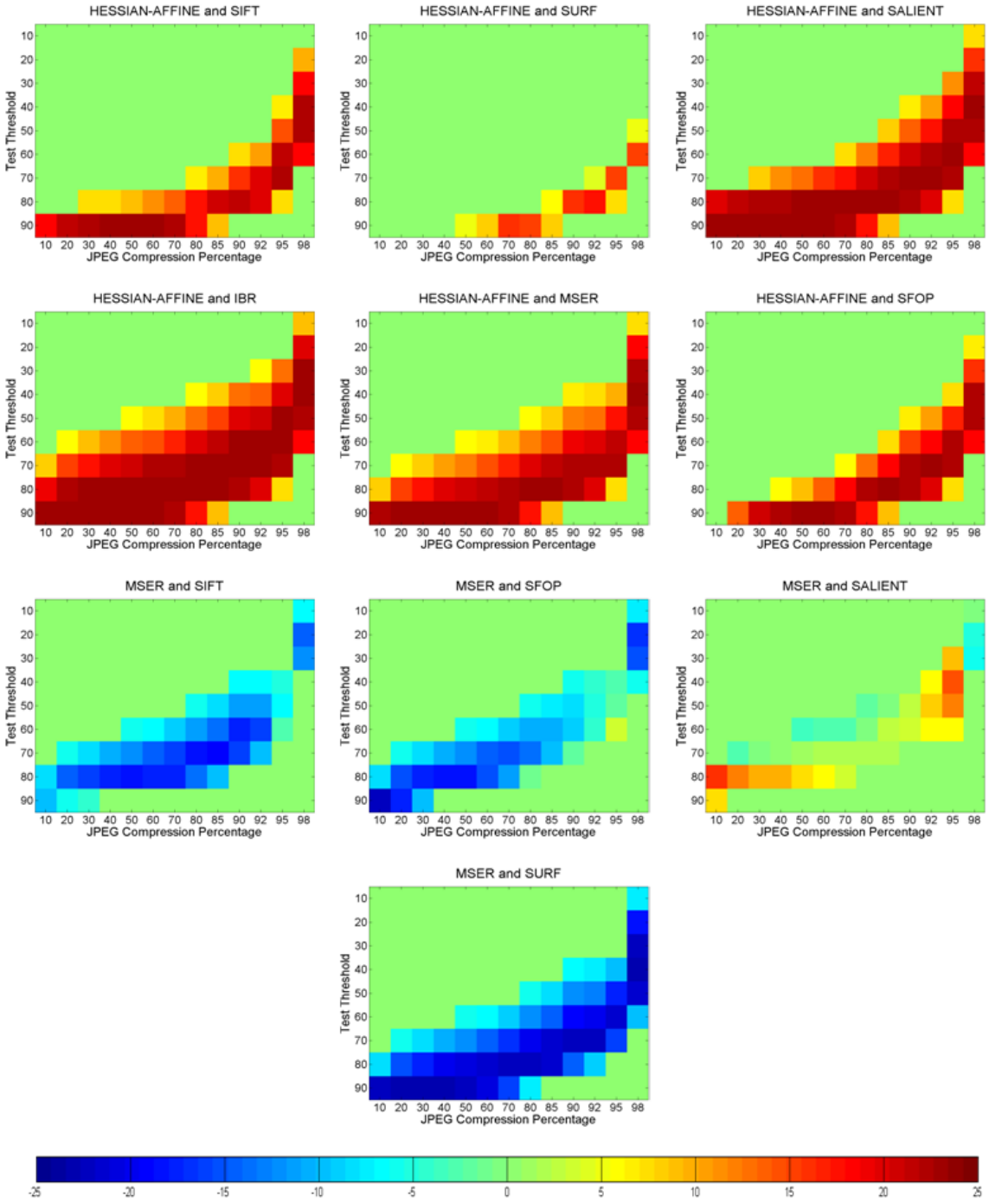

Figure 4-17: JPEG database results for Hessian-Affine and MSER with the other detectors showing Z-scores obtained utilizing the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas the negative values show the converse

4.4 Results for Blur

This section presents results for some of these feature detectors utilizing the proposed generic framework under changes in blur. A newly acquired database is employed to obtain these results which establish the upper and

lower performance bounds of the detectors and find statistically significant performance differences between them.

4.4.1 Blur Image Database

For investigating the behavior of local feature detectors utilizing the proposed framework under blur changes, a large database of images involving the same scenes as in the JPEG image database is presented with variations in the amount of blur. It should be noted that only two datasets of six images each, one containing a textured and the other a structured scene (Trees and Bikes datasets respectively), are used in [46] to determine the performance of six state-of-the-art local feature detectors. Instead, the presented database consists of 5390 images in total with 539 different planar scenes captured by the author. Both structured and textured real-world scenes are included to ensure that there is no bias shown towards any particular detector when determining the upper and lower performance bounds. Some images from the blur image database are shown in Figure 4-18.

Each image in the presented database consists of 717 x 1080 pixels. The amount of blur is varied in 10 discrete steps for each scene (10 x 539 = 5390). The database has been generated digitally utilizing MATLAB; the first image of every scene (having no blur) is convolved repeatedly with Gaussian blur kernels, having the same size as the image, with increasing standard deviations to produce a sequence of images with increasing amount of blur. More specifically, standard deviations ranging from 0.5 to 4.5 with a step size of 0.5 are used for the blur kernels. Since the increasing amount of blur does not cause any geometric transformation in the image with respect to the previous images in the same sequence, the ground truth homography which provides the image-to-image mapping for any two images of the same scene with different amounts of blur is simply a 3 x 3 identity matrix. Finally, in an effort to make the presented database a benchmark against which the future detectors can be examined, it has been made available at [182].

Figure 4-18: Some images from the Blur image database

4.4.2 Establishing Operating and Guarantee Regions

Figure 4-19 to Figure 4-28 present results for several state-of-the-art feature detectors utilizing the proposed generic framework which establish their *operating* and *guarantee* regions under changes in blur. Again, there is no other detailed work in the literature with which these results can be compared to determine the consistencies and contradictions. In [46], the authors had concluded on the basis of only two datasets that the detectors under examination are robust to changes in blur, as they featured almost horizontal repeatability curves. The results presented here are more comprehensive and largely contradict that perception. It should be noted that SIFT detects more than 20,000 features for some images in the blur image database which makes it very time-consuming to do such a detailed analysis for SIFT. In the case of JPEG image database, it took more than two months to obtain results on HP ProLiant DL380 G7 system with Intel Xeon 5600 series processors. Therefore, results for SIFT are not provided in this section.

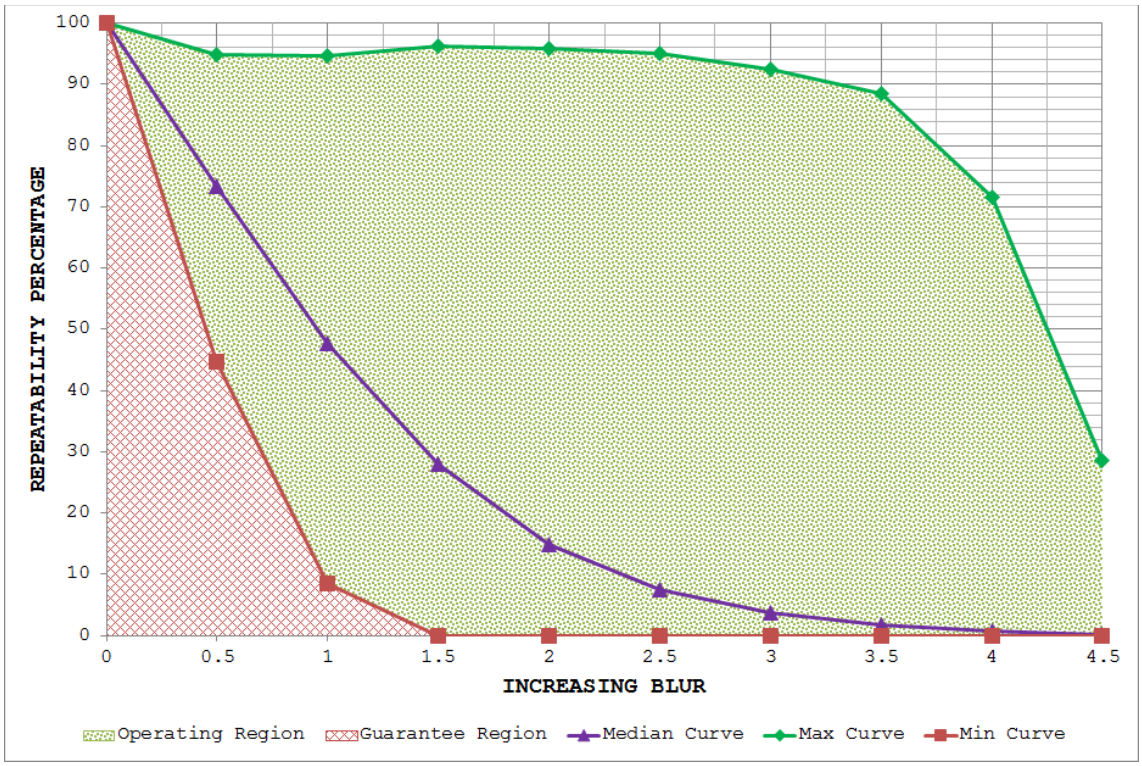

Figure 4-19: Blur database results for MSER utilizing the proposed framework

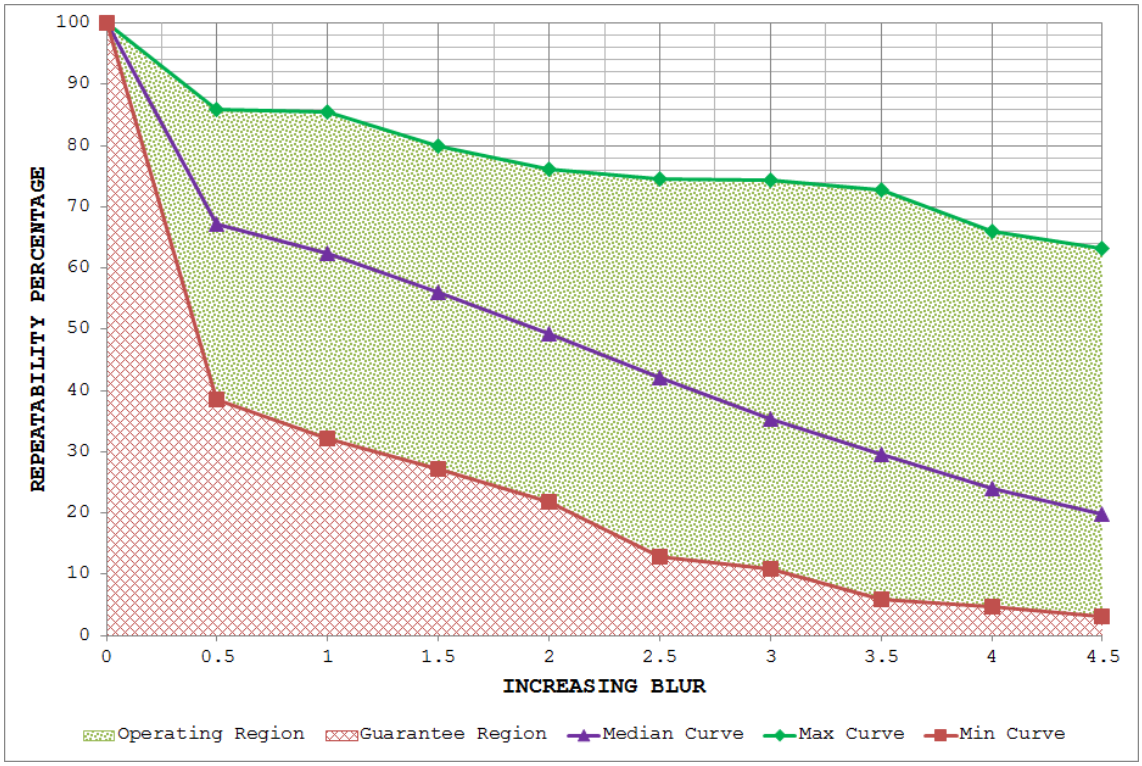

Figure 4-20: Blur database results for IBR utilizing the proposed framework

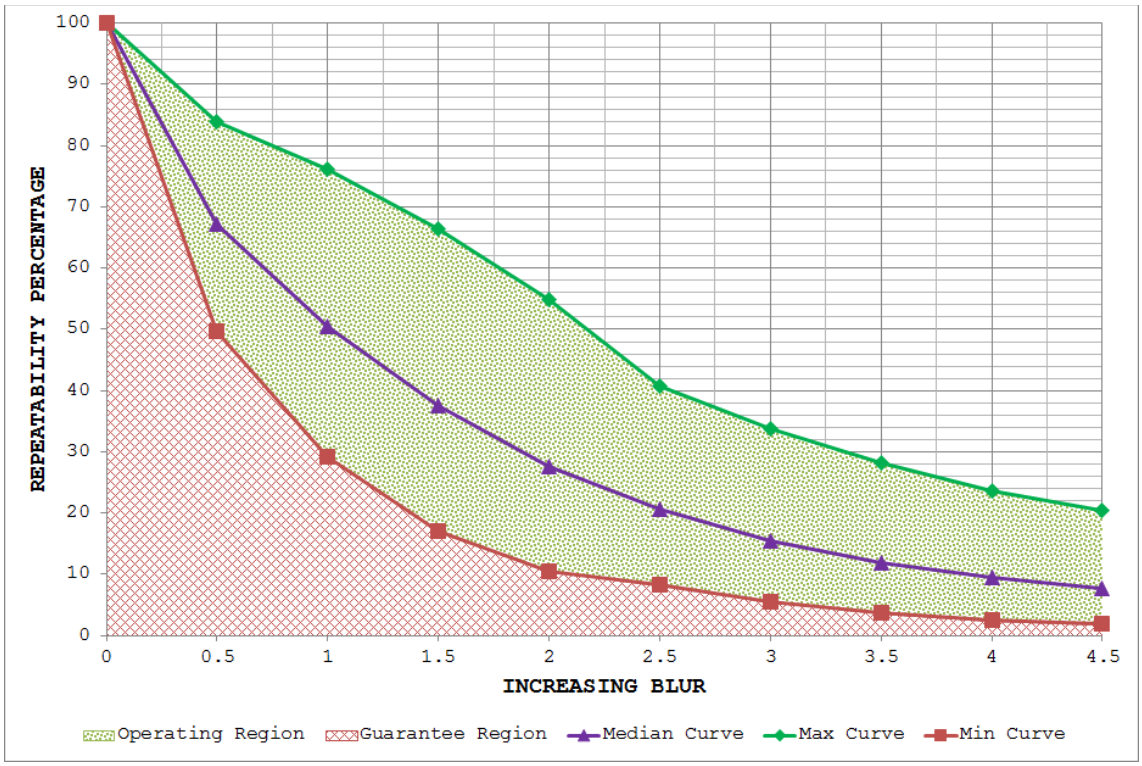

Figure 4-21: Blur database results for Salient detector utilizing the proposed framework

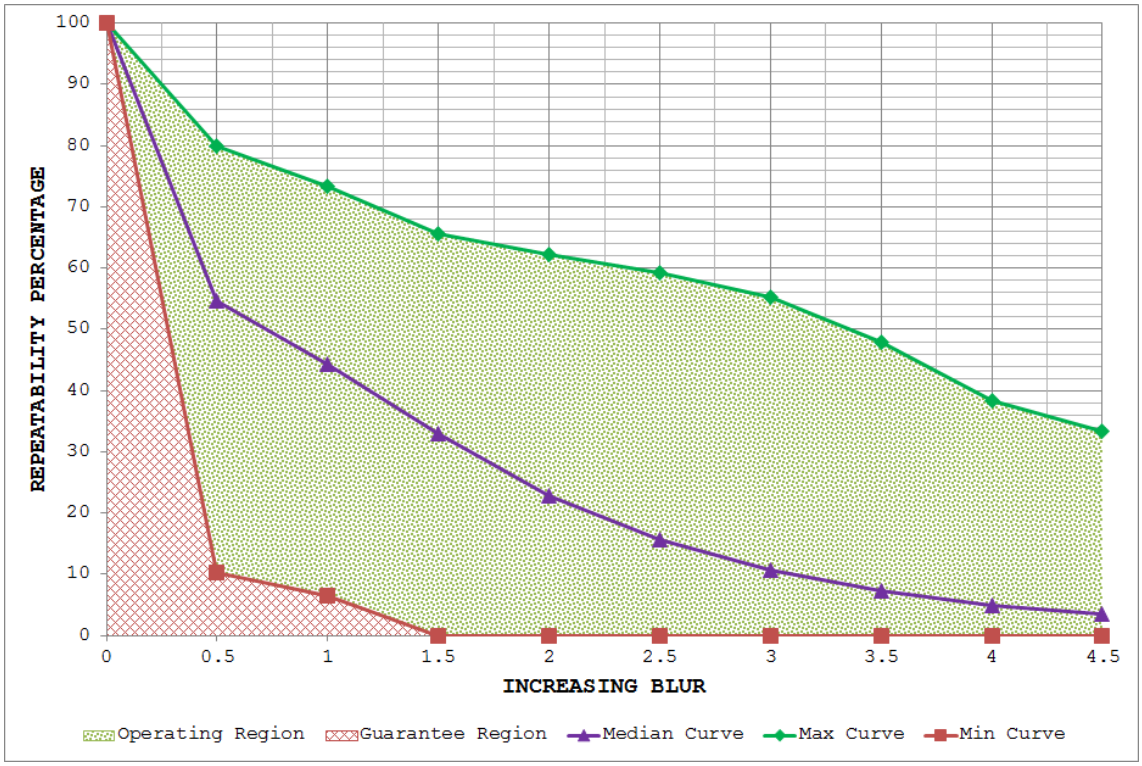

Figure 4-22: Blur database results for EBR utilizing the proposed framework

The results for MSER and IBR are shown in Figure 4-19 and Figure 4-20 respectively. It is evident that the two detectors undergo a decline in performance with increasing amount of blur. Both MSER and IBR are segmentation-based detectors but it is interesting to note that the *guarantee region* of IBR is much wider than that of MSER. Moreover, the *operating region* of MSER is large, indicating that the detector is unstable; it may provide high repeatability scores for some particular images yet may fare poorly for others. Such unpredictable behavior is not suitable from the vision systems design viewpoint.

Figure 4-21 depicts the *operating* and *guarantee* regions for Salient. There is continuous degradation in the performance of Salient with increasing blur. However, the detector appears more stable than MSER and IBR, as indicated by a narrow *operating region*. The results for EBR are shown in Figure 4-22. The performance of EBR depends largely upon the image content as it may achieve good repeatability scores for some image while its performance may go to zero for others. A large operating region for EBR points to its unstable behavior under changes in the amount of blur.

The *operating* and *guarantee* regions for SURF are shown in Figure 4-23. It is clear that the *operating region* of SURF grows rapidly with increasing blur, thus indicating unpredictable behavior of the detector. Depending upon the image content, SURF may fail to provide repeatable features in the presence of increasing blur (see the min curve in Figure 4-23). Conversely, SFOP shows comparatively good performance with a large *guarantee region* and a narrow *operating region* (see Figure 4-24). It seems quite stable under increasing blur and the max and min curves are also fairly smooth, indicating a gradual degradation in performance. Figure 4-25 to Figure 4-28 depict results for Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine respectively. The *operating* and *guarantee* regions of these four detectors are similar, although Hessian-based detectors appear better than Harris-based ones. For small amounts of blur, the detectors demonstrate good performance but may fare poorly in the presence of increasing blur.

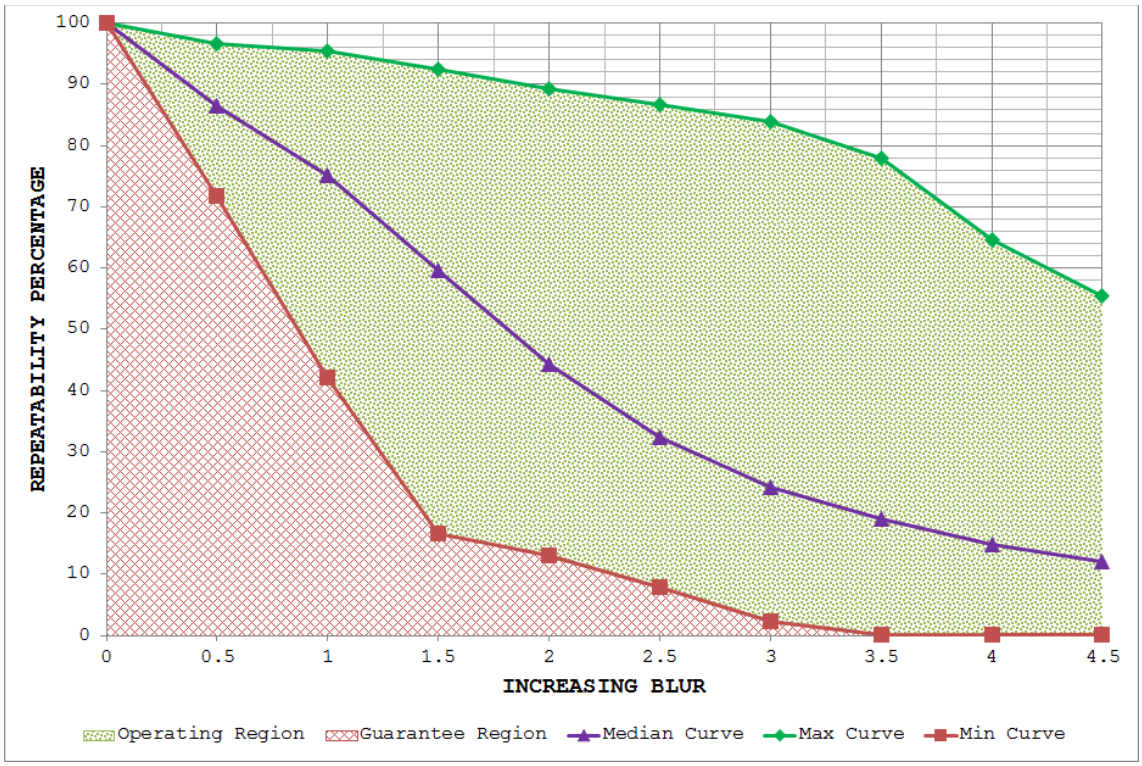

Figure 4-23: Blur database results for SURF detector utilizing the proposed framework

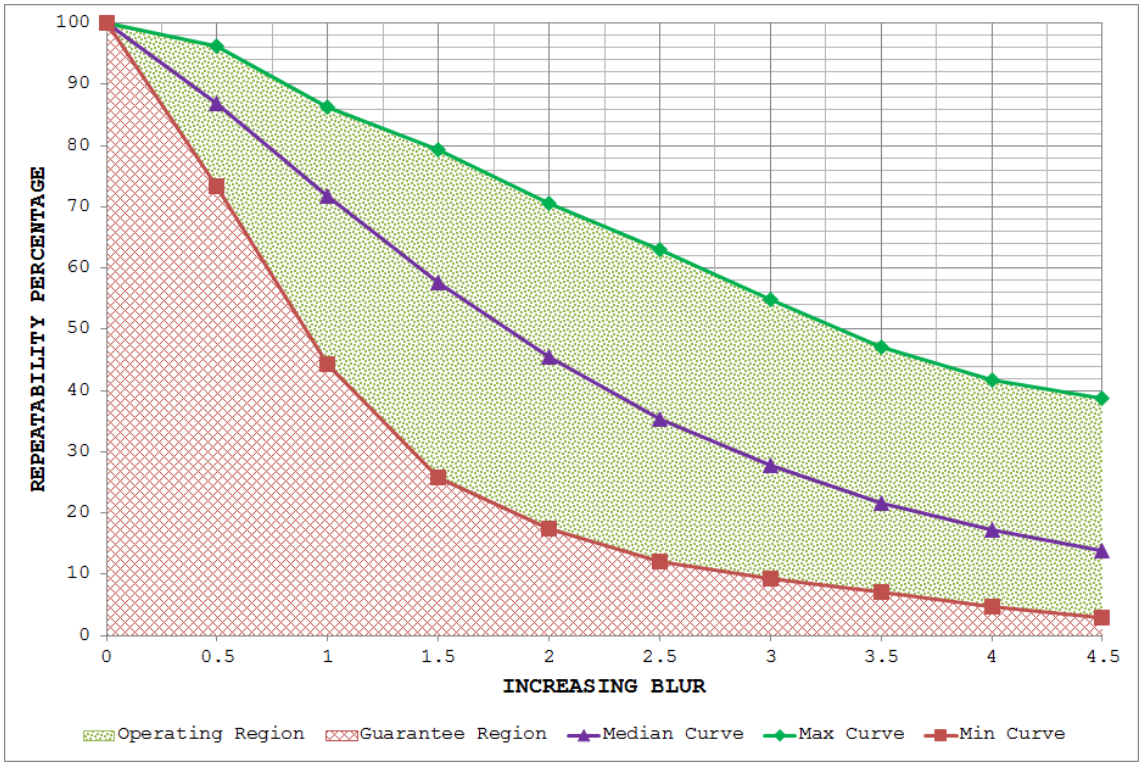

Figure 4-24: Blur database results for SFOP utilizing the proposed framework

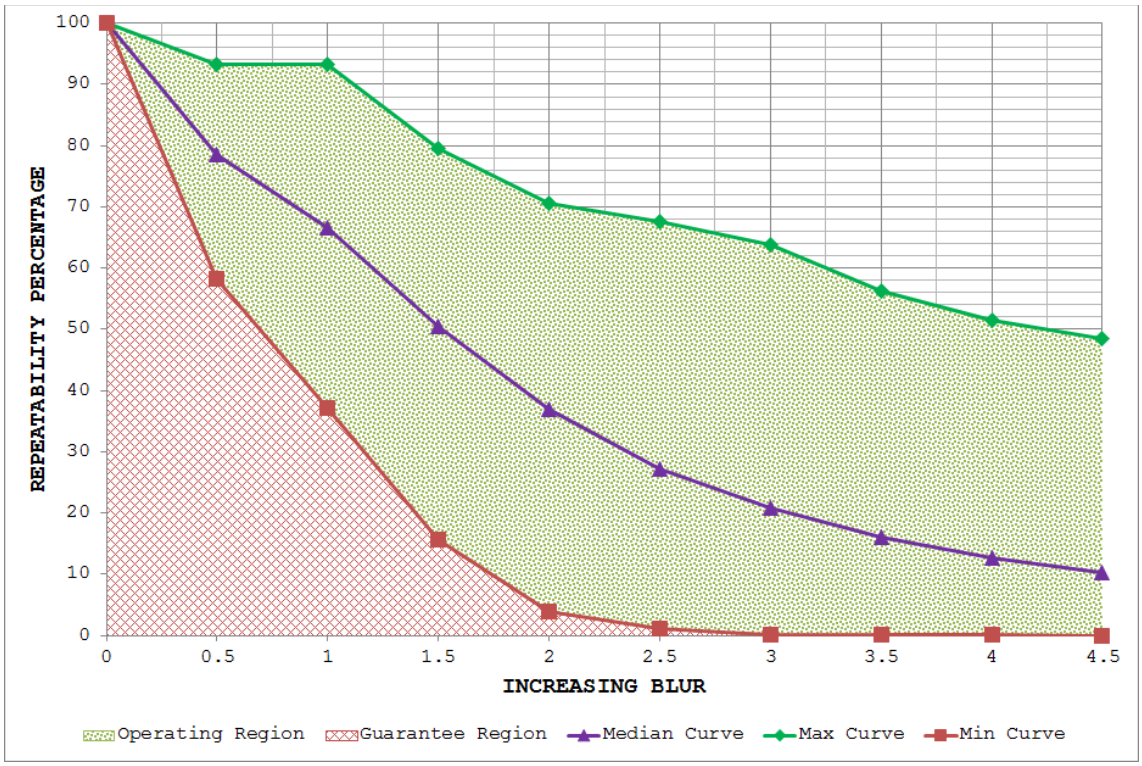

Figure 4-25: Blur database results for Harris-Laplace utilizing the proposed framework

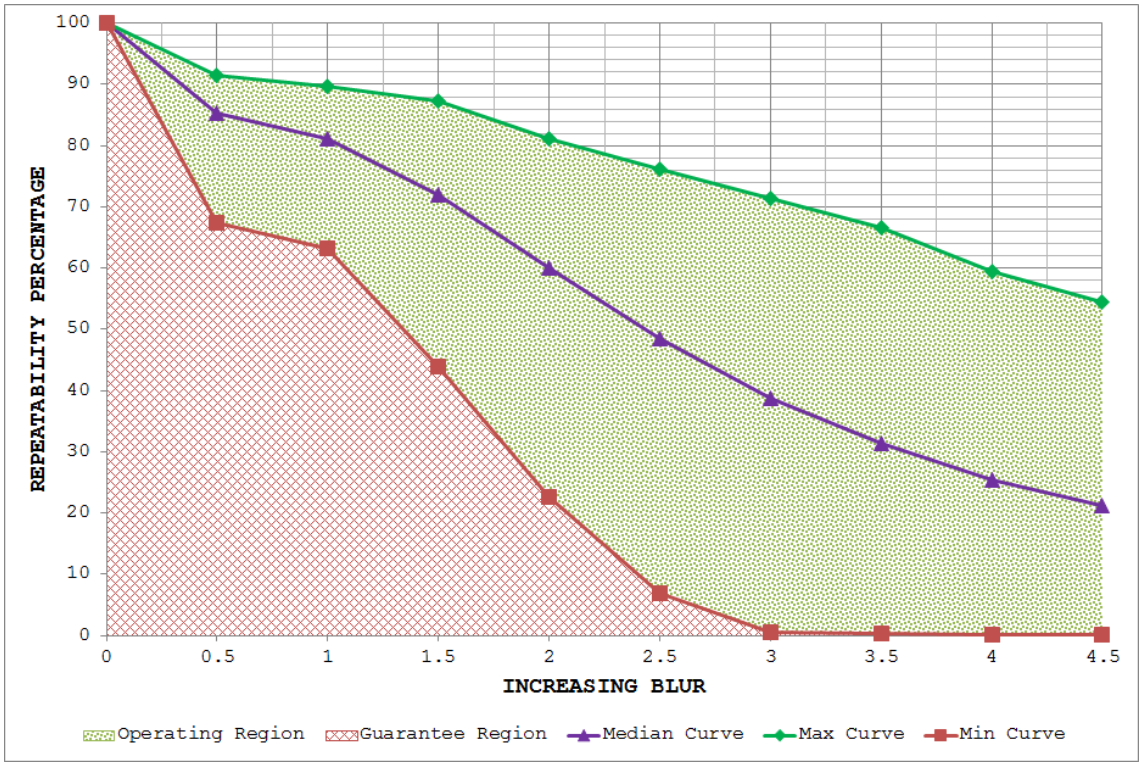

Figure 4-26: Blur database results for Hessian-Laplace utilizing the proposed framework

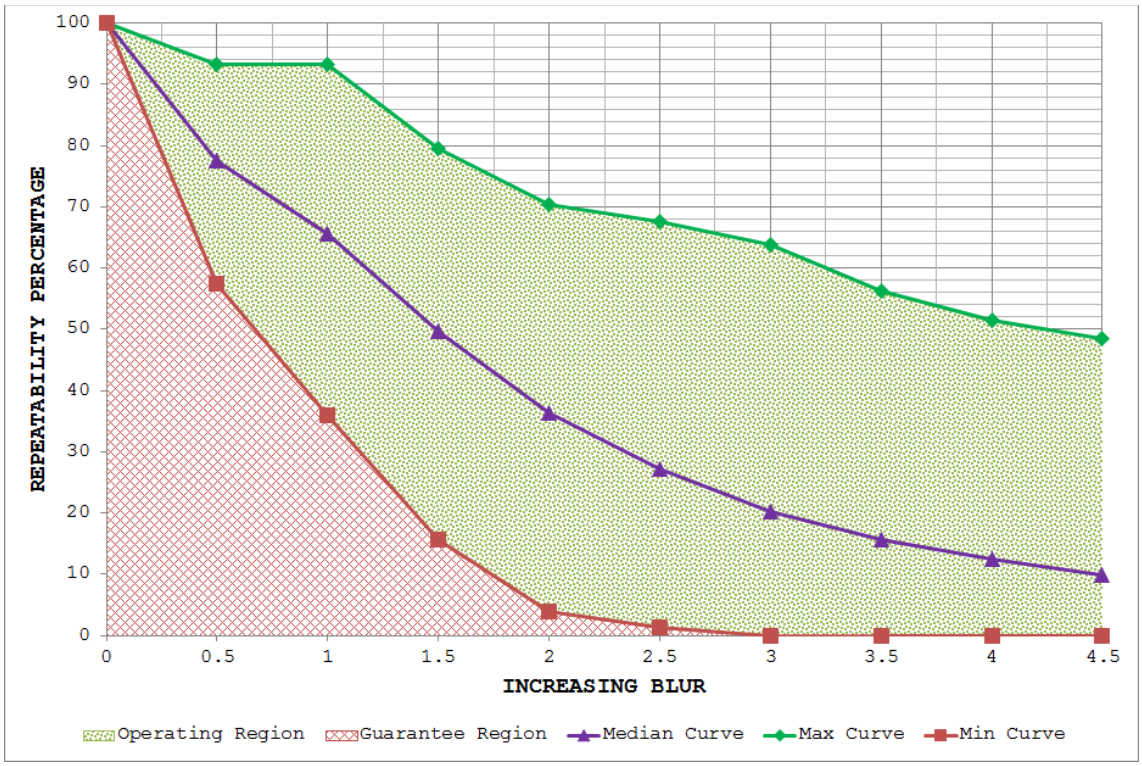

Figure 4-27: Blur database results for Harris-Affine utilizing the proposed framework

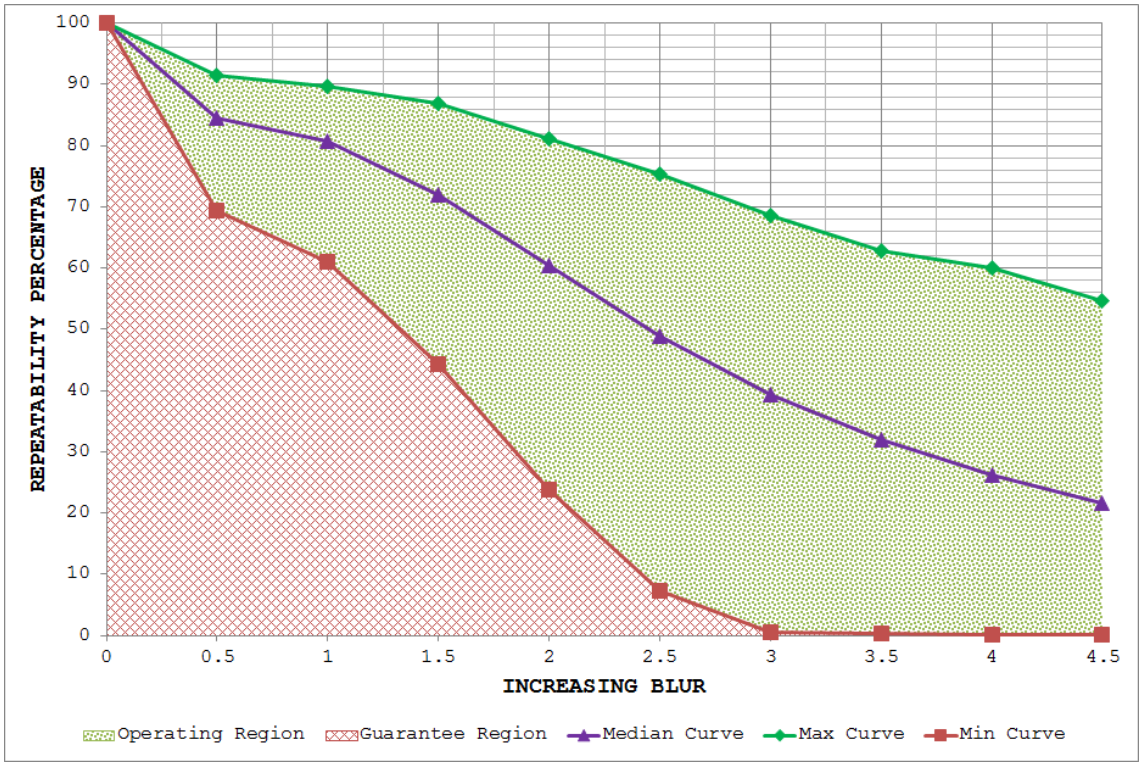

Figure 4-28: Blur database results for Hessian-Affine utilizing the proposed framework

4.4.3 Identifying Statistically Significant Performance Differences

Figure 4-29 to Figure 4-31 depict the results for statistical performance comparison of state-of-the-art feature detectors utilizing the proposed framework under changes in blur. As in the figures of Section 4.3.3, color coding in Figure 4-29 to Figure 4-31 indicates the *Z-scores* obtained as a function of image transformation amount and McNemar's test threshold when one detector is compared with another. The same sign convention has been used to distinguish the detector with the better performance of the two examined: a positive *Z-score* shows that the first detector is better than the second whereas a negative value indicates the converse.

It is clear from Figure 4-29 that the performance differences between Hessian-Laplace and other detectors, except Hessian-Affine, are statistically significant for most test thresholds and blur amounts, with Hessian-Laplace turning out to be the better detector of the two compared. Harris-Laplace performs better than MSER and Salient (see Figure 4-29) but is dominated by Hessian-Affine, IBR, SFOP and SURF for most test thresholds and blur amounts. In Figure 4-30, it is evident that EBR is comprehensively out-performed by all other detectors but MSER. Harris-Affine and Harris-Laplace show largely similar performances (see Figure 4-30). The performance of Harris-Affine is better than MSER and Salient in Figure 4-30, while large negative *Z-scores* show the supremacy of Hessian-Affine, IBR, SFOP and SURF over Harris-Affine. Like Hessian-Laplace, Hessian-Affine out-performs IBR, MSER, Salient, SFOP and SURF in Figure 4-31. The performance differences between IBR and MSER are statistically significant, with IBR appearing the better of the two. IBR also performs better than Salient. When IBR is compared with SURF and SFOP, both positive and negative *Z-scores* are obtained for different test thresholds and blur amounts (see Figure 4-31). Also, MSER is dominated by Salient, SURF and SFOP, whereas Salient is out-performed by SFOP and SURF.

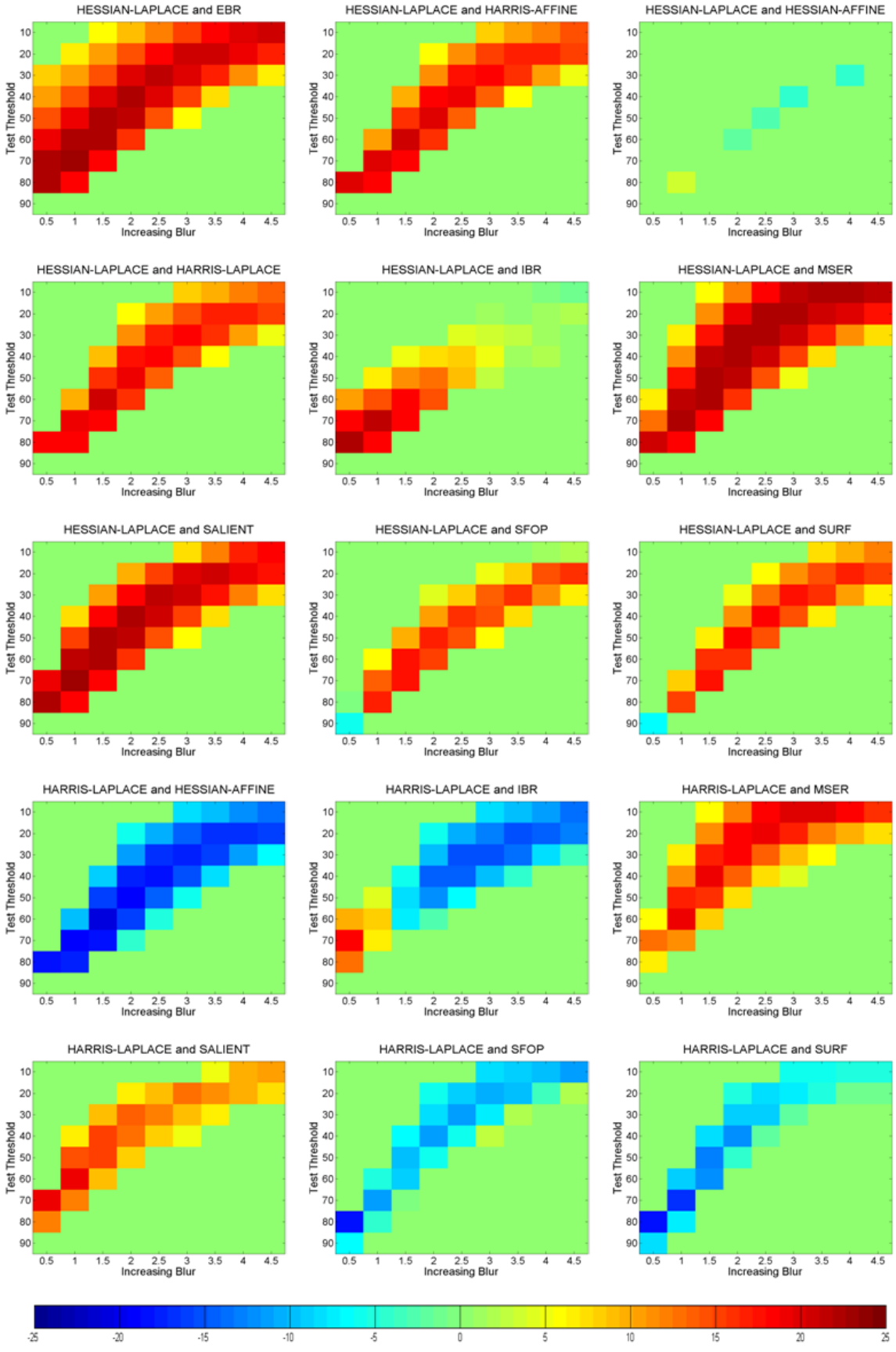

Figure 4-29: Blur database results for Harris-Laplace and Hessian-Laplace with other detectors showing Z-scores obtained using the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas negative values show the converse

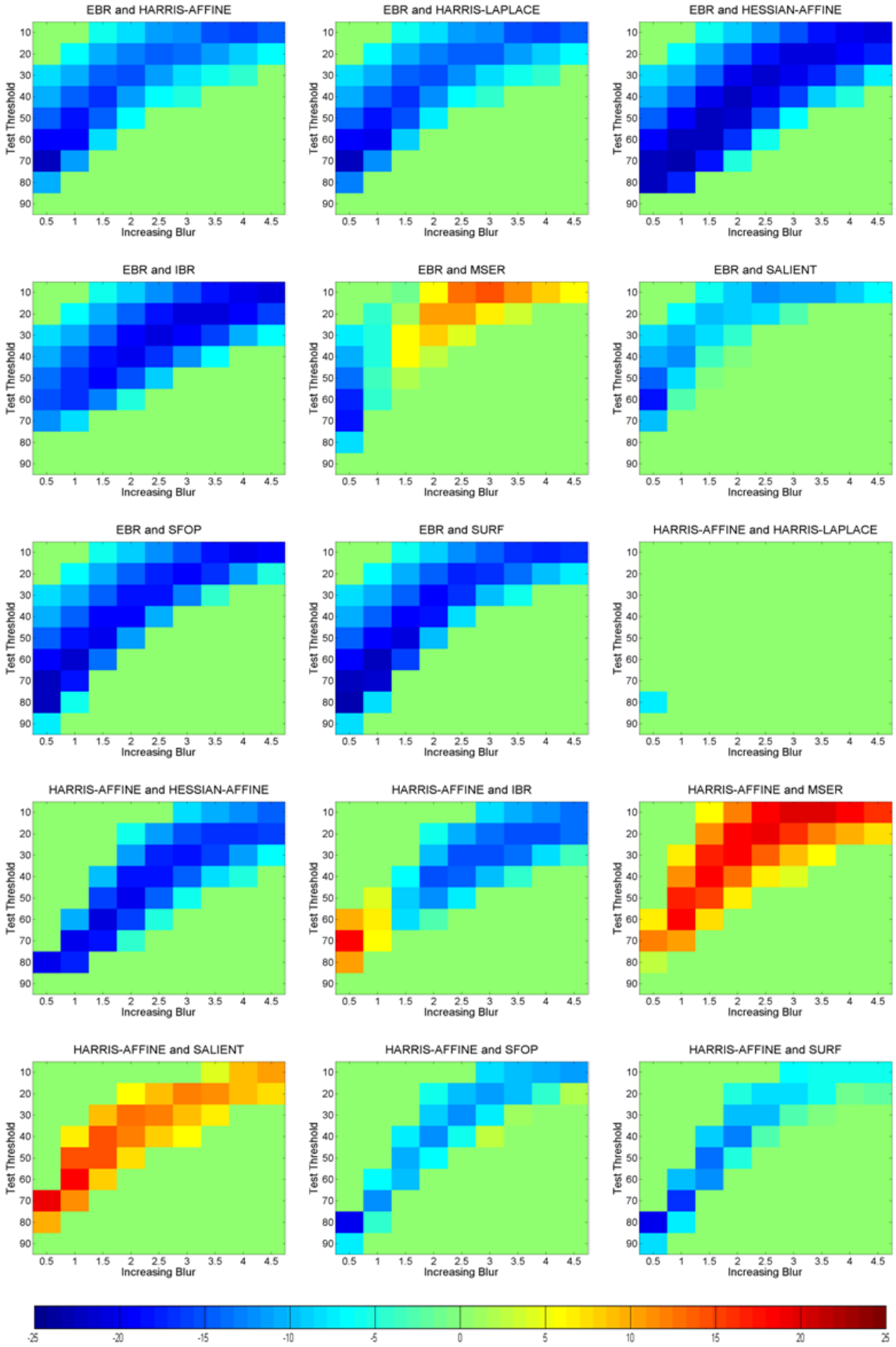

Figure 4-30: Blur database results for EBR and Harris-Affine with the other detectors showing Z-scores obtained using the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas the negative values show the converse

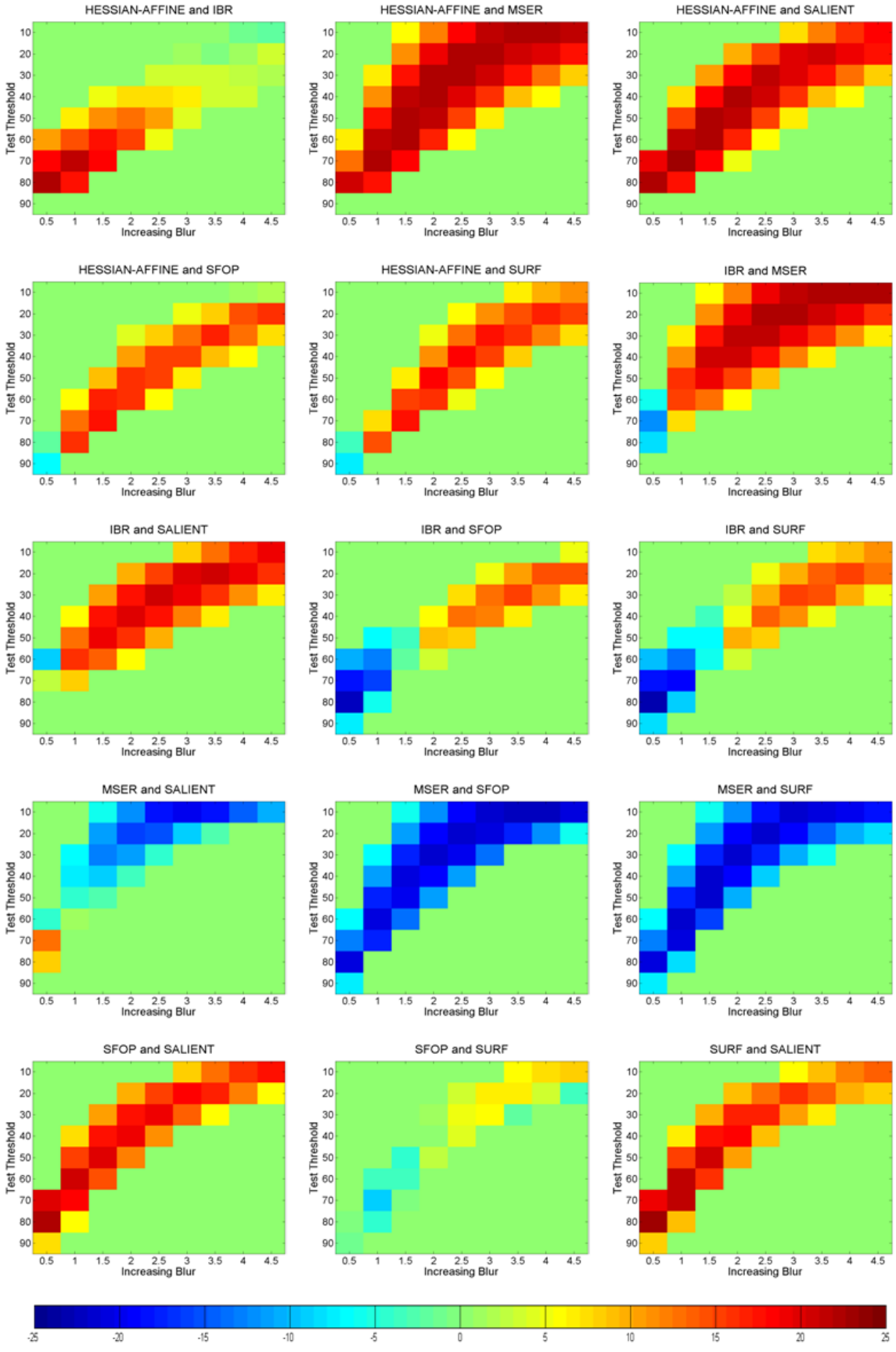

Figure 4-31: Blur database results for Hessian-Affine, IBR, MSER, SFOP and SURF with other detectors showing Z-scores obtained using proposed framework; positive Z-scores show that the first detector is better than the second whereas negative values show the converse

4.5 Results for Uniform Light Changes

This section presents results for the state-of-the-art feature detectors under uniform changes in illumination. These results determine the *operating* and *guarantee* regions of the detectors by employing a large image database and identify statistical performance differences between them under uniform changes in light.

4.5.1 Light Image Database

Among the Oxford datasets [50], only Leuven, consisting of a sequence of six images, involves uniform changes in light. In [177], a large image database is presented to investigate the effect of light direction on the performance of feature detectors. However, the total number of scenes that have been used in that database is only 60. Also, the scenes employed are not real-world scenes but are captured in a highly controlled environment. A large image database is thus presented in this section to investigate the behavior of local feature detectors under uniform changes in illumination by employing the framework proposed in Section 4.2. The database consists of 7546 images, involving the same 539 scenes as in the JPEG and blur image databases, with variation in illumination. Both structured and textured real-world scenes are included. The number of scenes for the presented database is nearly 9 times that of what is used in [177]. Some images from the light image database are shown in Figure 4-32.

Each image in the database consists of 717 x 1080 pixels. For every scene, the brightness level is decreased in 14 discrete steps from 0% to 90% (14 x 539 = 7546). The database has been generated digitally using MATLAB by varying the image brightness level. The ground truth homography that relates any two images of the same scene with different light conditions in the presented database is a 3 x 3 identity matrix as uniform changes in light do not result in any geometric transformation. The image database has been made available at [183] to facilitate future research.

Figure 4-32: Some images from the Light image database

4.5.2 Establishing Operating and Guarantee Regions

Figure 4-33 to Figure 4-42 depict the upper and lower performance bounds of several state-of-the-art feature detectors under uniform changes in light utilizing the proposed generic framework. Again, the results of SIFT are not provided as it detects a large number of features in some images of the database (in excess of 20,000), making the computation time prohibitively large for such a detailed analysis.

In [46], the authors have concluded that the six detectors under study are highly robust to uniform variations in illumination. As mentioned earlier, this deduction is based on a single dataset (Leuven [50]). The results presented here largely contradict those findings, showing that there is a rapid decline in the performance in the presence of uniform light changes. A similar performance degradation effect is observed in [177] while studying the behavior of feature detectors under changes in light direction.

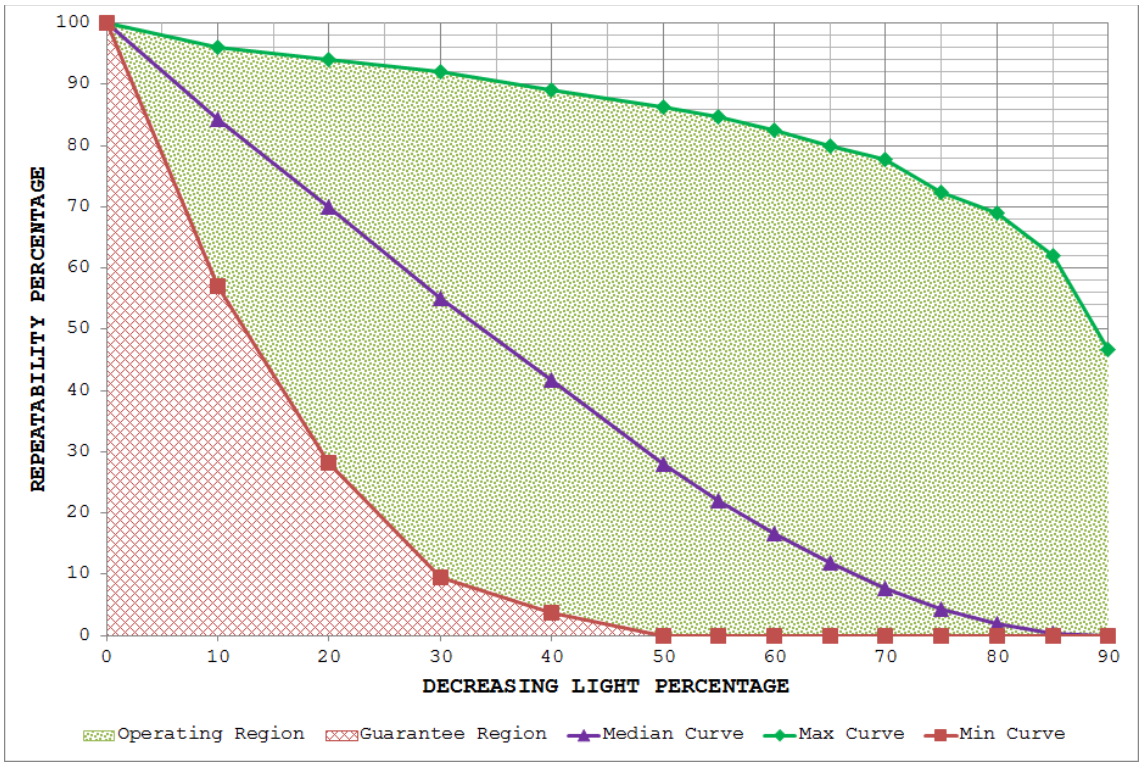

Figure 4-33: Light database results for MSER utilizing the proposed framework

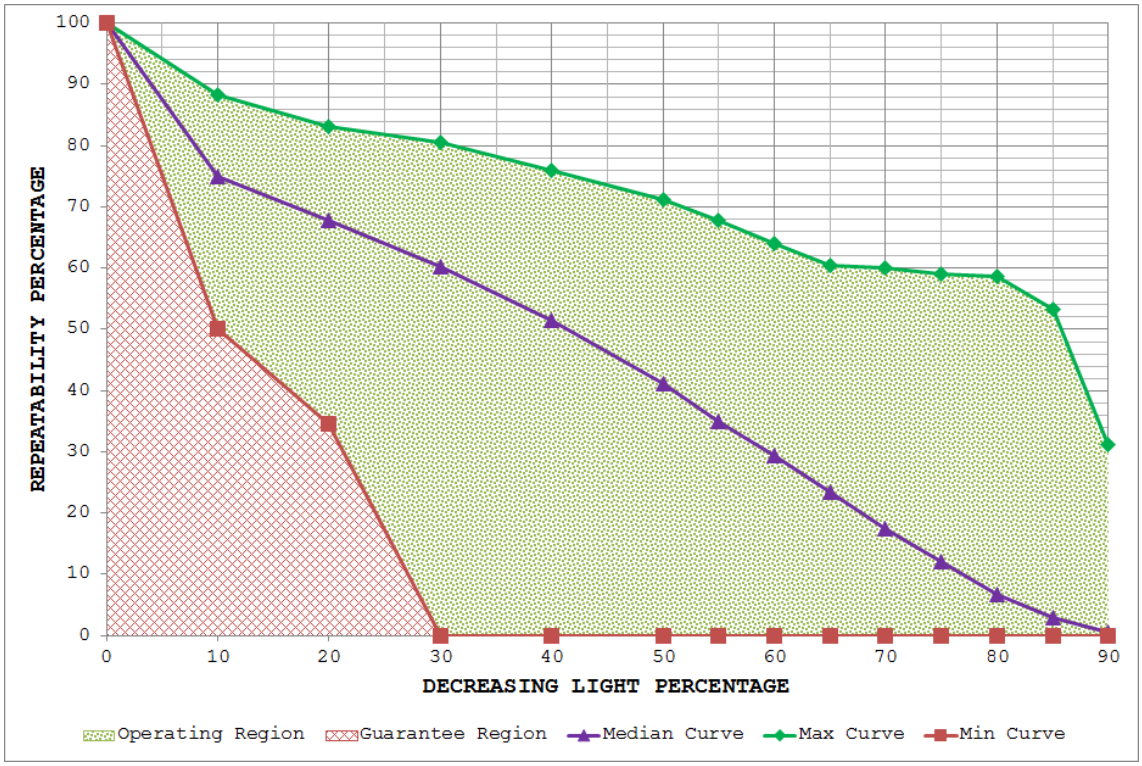

Figure 4-34: Light database results for IBR utilizing the proposed framework

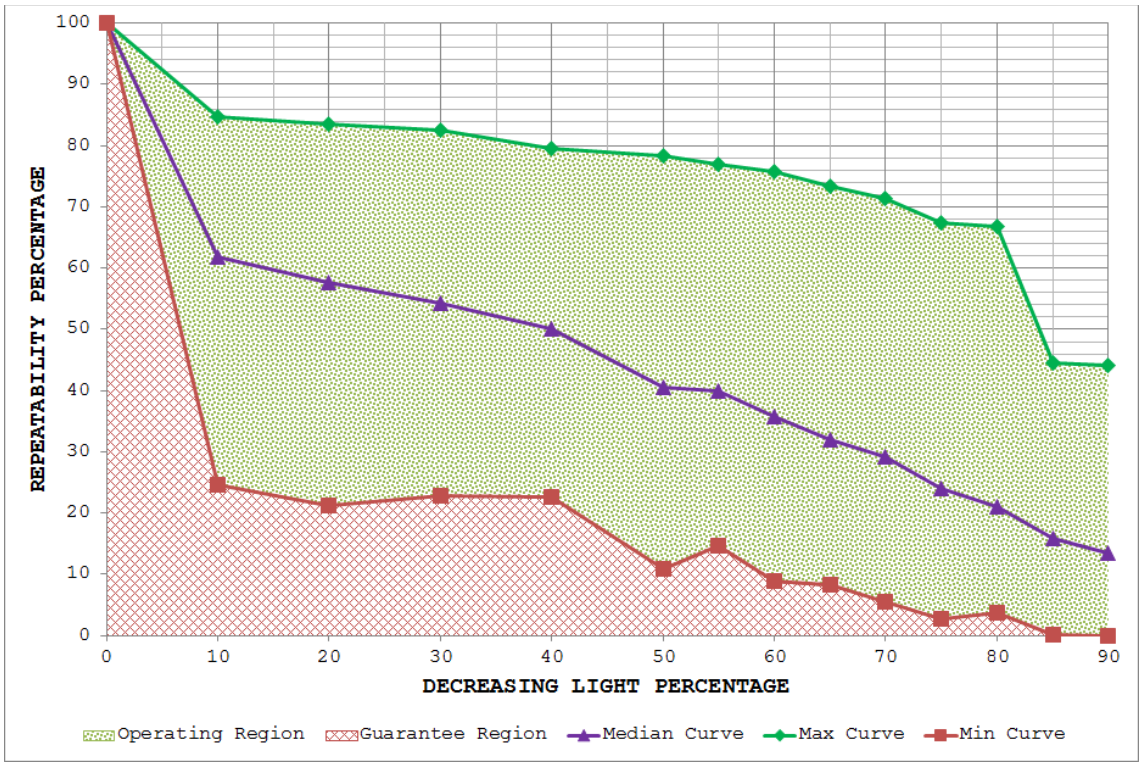

Figure 4-35: Light database results for Salient detector utilizing the proposed framework

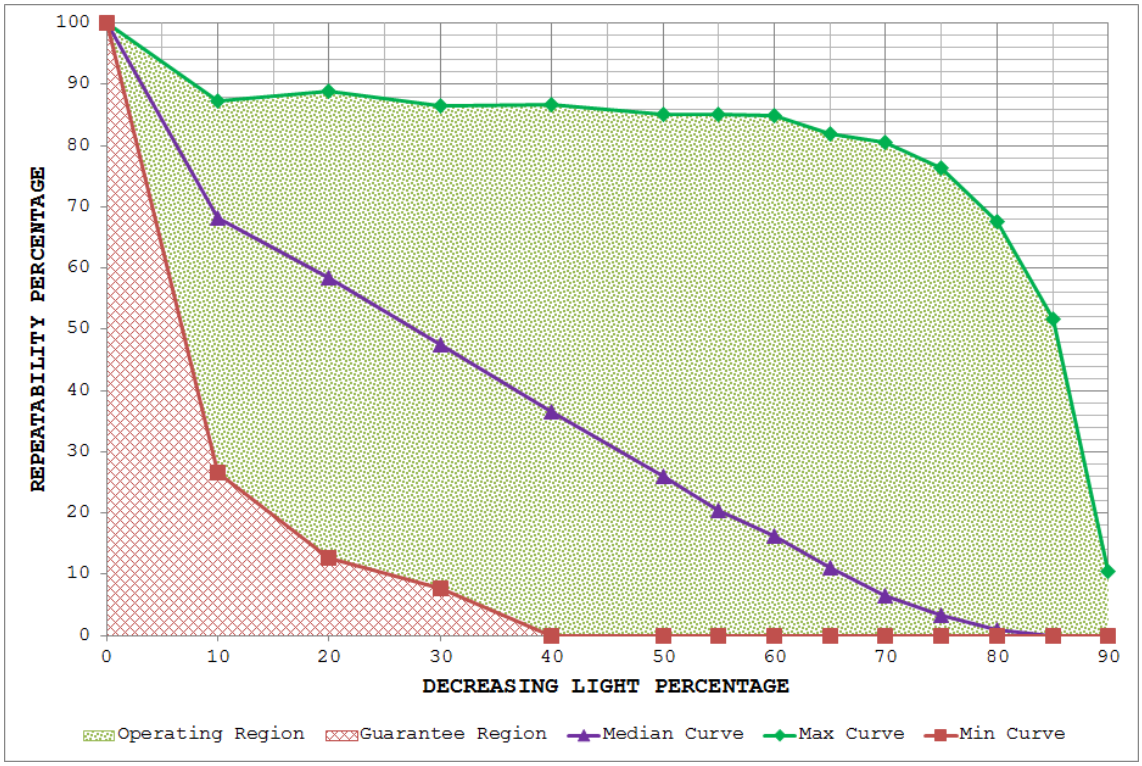

Figure 4-36: Light database results for EBR utilizing the proposed framework

Figure 4-33 and Figure 4-34 show the results for the two segmentation-based detectors, MSER and IBR, respectively. It is evident that both the detectors have very large *operating* regions which indicate their unstable behavior in the presence of decreasing light. It is interesting to note that the min curve of MSER, the detector which is identified as the best for this specific image transformation in [46], reaches zero for only 50% uniform decrease in light (see Figure 4-33). MSER and IBR do not seem suitable for vision systems expecting more than 10% uniform decrease in light.

The *operating* and *guarantee* regions for Salient and EBR are depicted in Figure 4-35 and Figure 4-36 respectively. As with MSER and IBR, both Salient and EBR have large *operating regions,* indicating that they may achieve high repeatability values for some particular images yet may fare poorly for some other images in the presence of uniform changes in illumination; such unpredictable behavior is not desirable from a vision systems design perspective. There is a rapid decline in the performance of SURF with decreasing light (see Figure 4-37). SFOP seems to perform much better as indicated by its wide *guarantee region*. However, its *operating region* shows that the performance of the detector may vary between high and low values of repeatability.

The results for Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine are depicted in Figure 4-39 to Figure 4-42 respectively. It is clear that all these detectors undergo a quick degradation in performance with decreasing light. The min curves for these detectors show that they hardly manage to achieve a repeatability score of 10-15% in the presence of only 20% uniform decrease in light (see Figure 4-39 – Figure 4-42). The *operating* regions of these detectors are large and their *guarantee* regions are narrow, meaning that they may achieve high repeatability scores for some images but may fare poorly for others. This large variation in performance for the same amount of image transformation is not desirable from a vision systems design perspective.

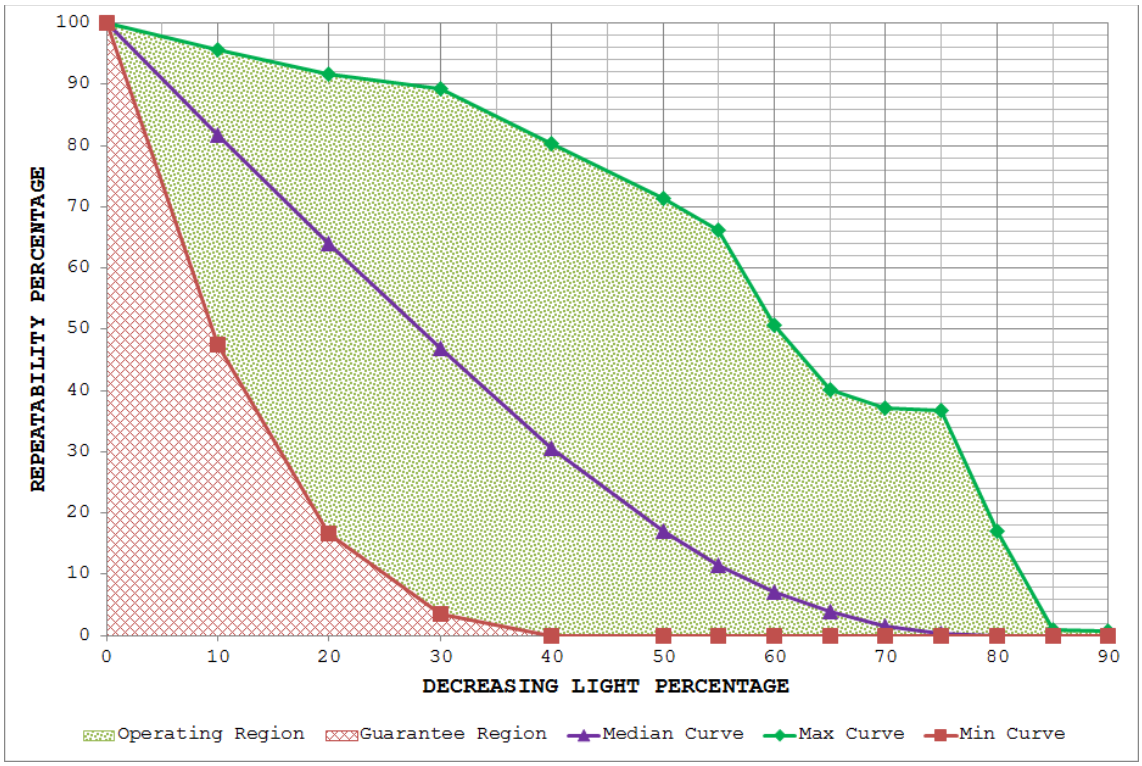

Figure 4-37: Light database results for SURF detector utilizing the proposed framework

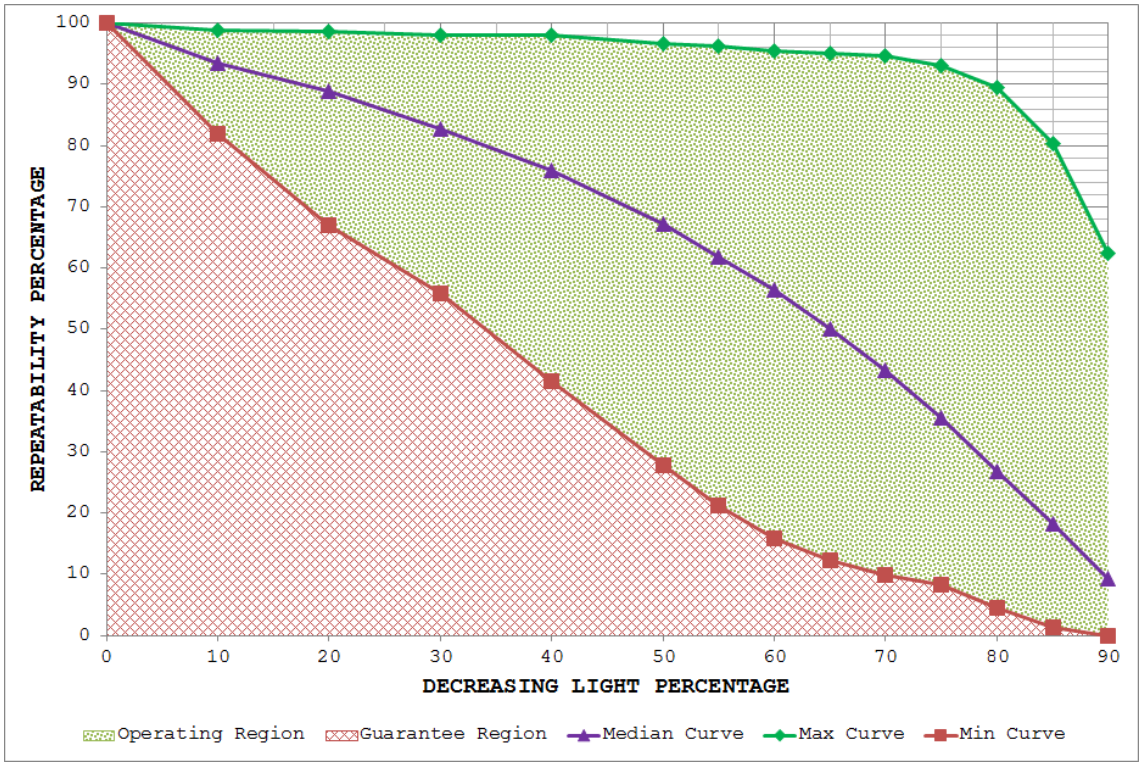

Figure 4-38: Light database results for SFOP utilizing the proposed framework

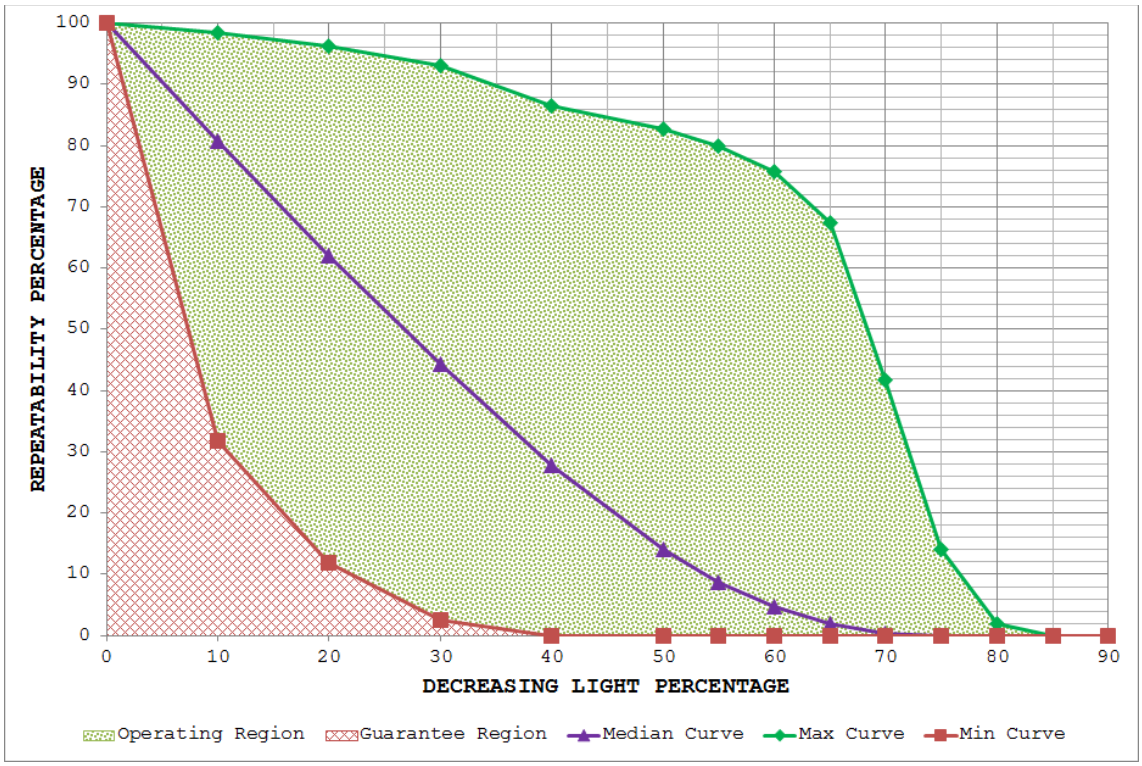

Figure 4-39: Light database results for Harris-Laplace utilizing the proposed framework

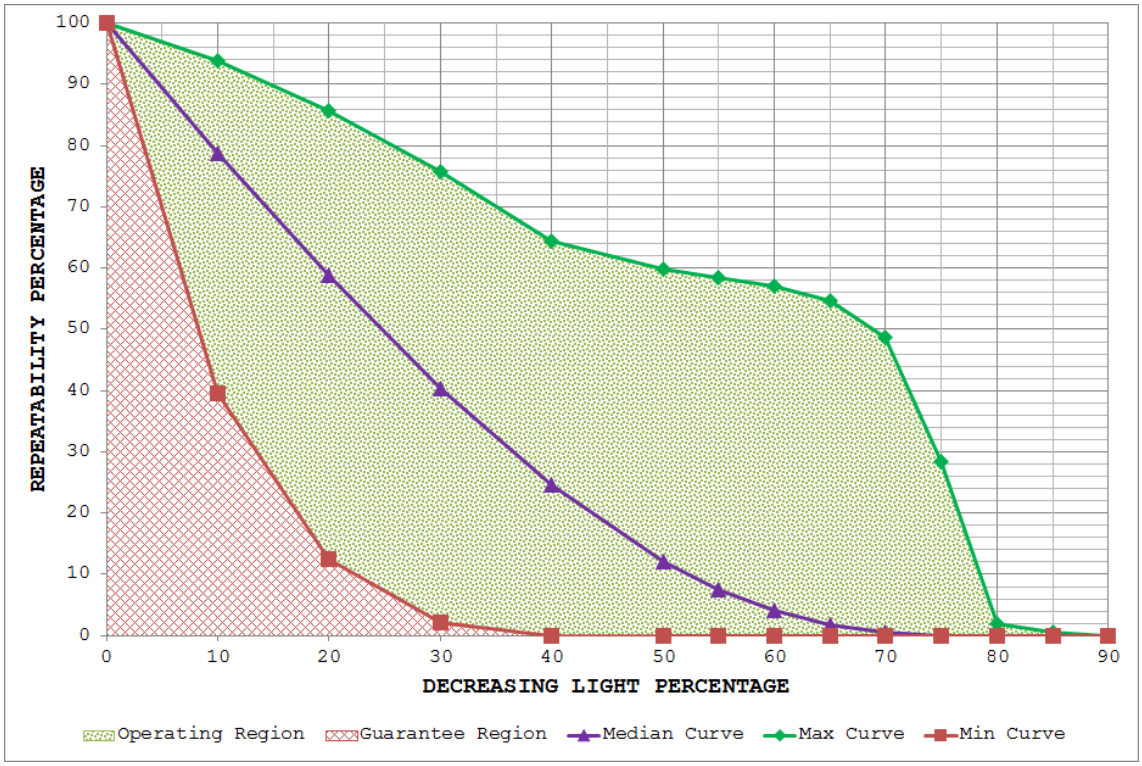

Figure 4-40: Light database results for Hessian-Laplace utilizing the proposed framework

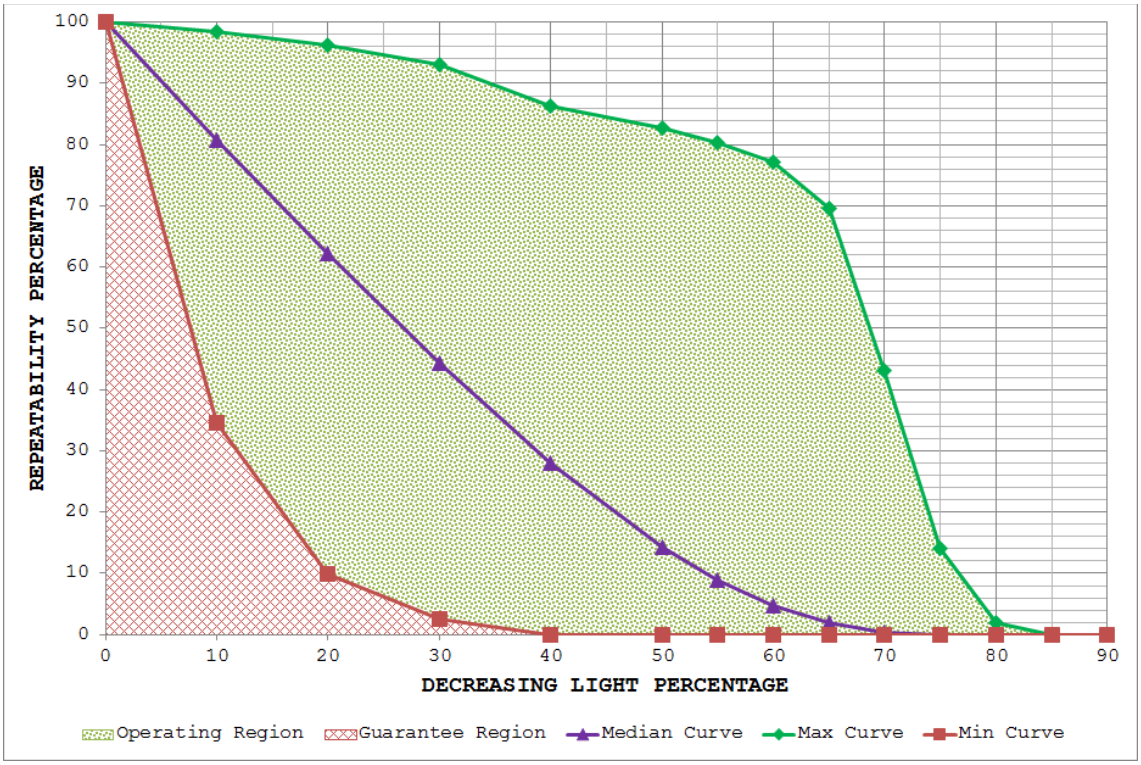

Figure 4-41: Light database results for Harris-Affine utilizing the proposed framework

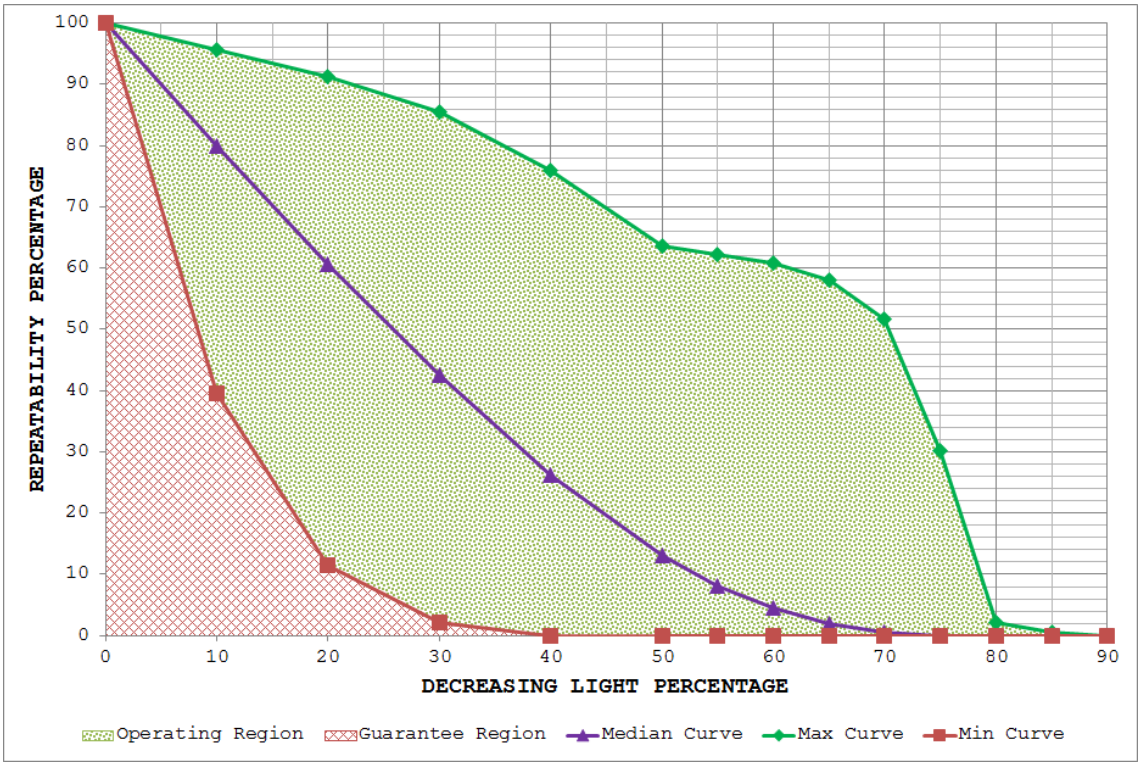

Figure 4-42: Light database results for Hessian-Affine utilizing the proposed framework

4.5.3 Identifying Statistically Significant Performance Differences

In an effort to find the statistically significant performance differences between these state-of-the-art feature detectors, results are presented in Figure 4-43 to Figure 4-45 utilizing the proposed framework. Again, the color coding in these figures indicate the *Z-scores* obtained as a function of image transformation amount and McNemar's test threshold when one detector is compared with another detector. The same sign convention has been used to distinguish the detector with the better performance: a positive *Z-score* shows that the first detector is better than the second and vice versa.

Figure 4-43 shows the performance differences of Harris-Laplace and Hessian-Laplace with the other detectors. It appears that nearly all detectors, including EBR, perform better than Hessian-Laplace for most test thresholds and decreasing light percentages. Apart from Hessian-Laplace and Hessian-Affine, Harris-Laplace fails to out-perform other detectors and is particularly dominated by SFOP, Salient and IBR. EBR shows better performance when compared with Hessian-Affine, Harris-Laplace, Harris-Affine and SURF (see Figure 4-44). SFOP, Salient and IBR show supremacy over EBR, whereas MSER has largely similar performance to EBR for most test thresholds and decreasing light percentages. As with Harris-Laplace, Harris-Affine is also out-performed by SFOP, Salient, IBR and MSER in Figure 4-44.

While Hessian-Affine and SURF seem to have largely similar performances, Hessian-Affine fares poorly when compared with IBR, MSER, Salient and SFOP (see Figure 4-45). Of the two segmentation-based detectors, IBR seems the better. MSER and IBR out-perform SURF but are dominated by SFOP and Salient. The performance differences of SURF with SFOP, and Salient are also statistically significant for most test thresholds and decreasing light percentages, with SURF emerging as the worst of the detectors compared. SFOP also out-performs Salient comprehensively.

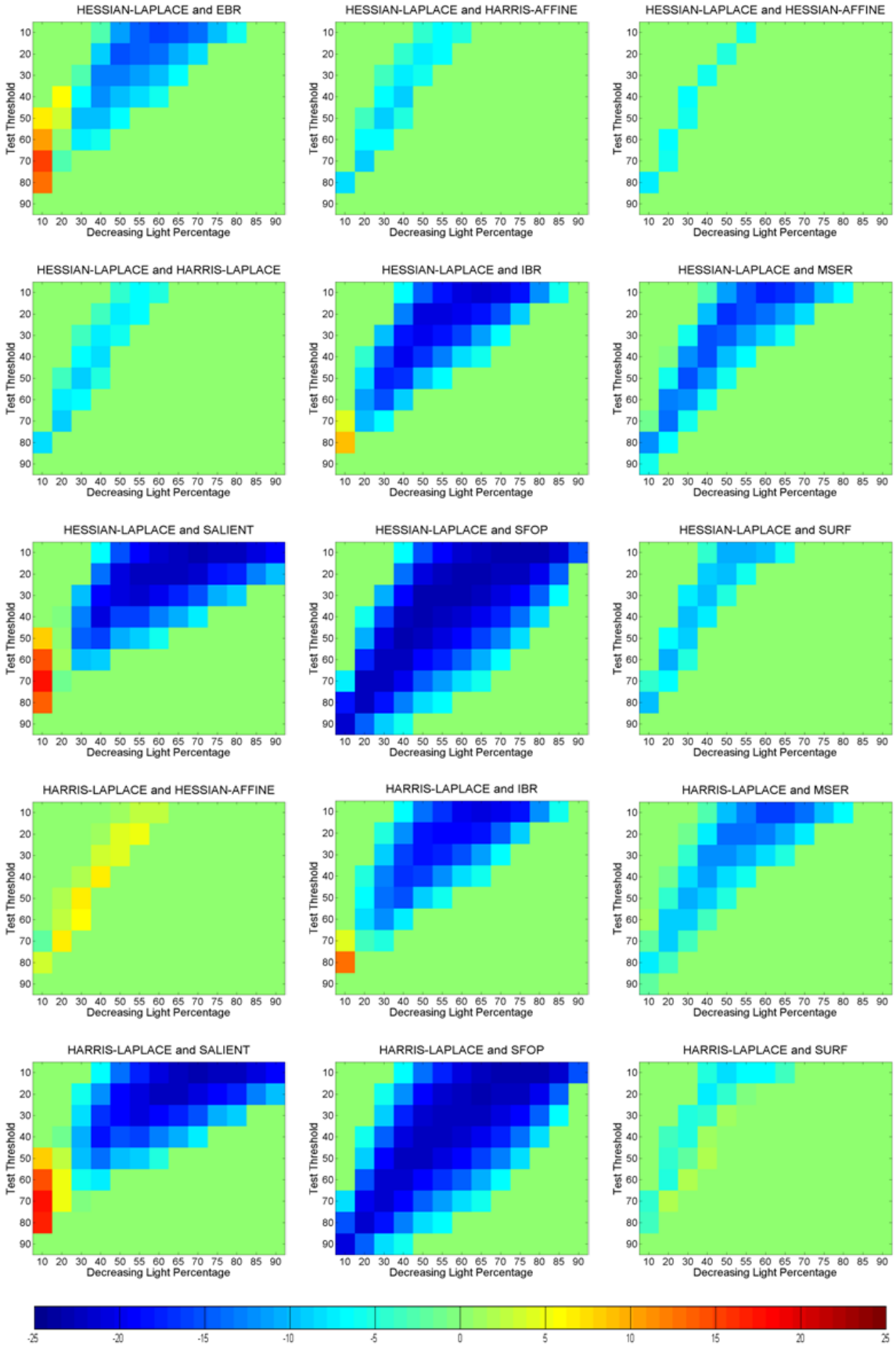

Figure 4-43: Light database results for Harris-Laplace and Hessian-Laplace with other detectors showing Z-scores obtained using the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas negative values show the converse

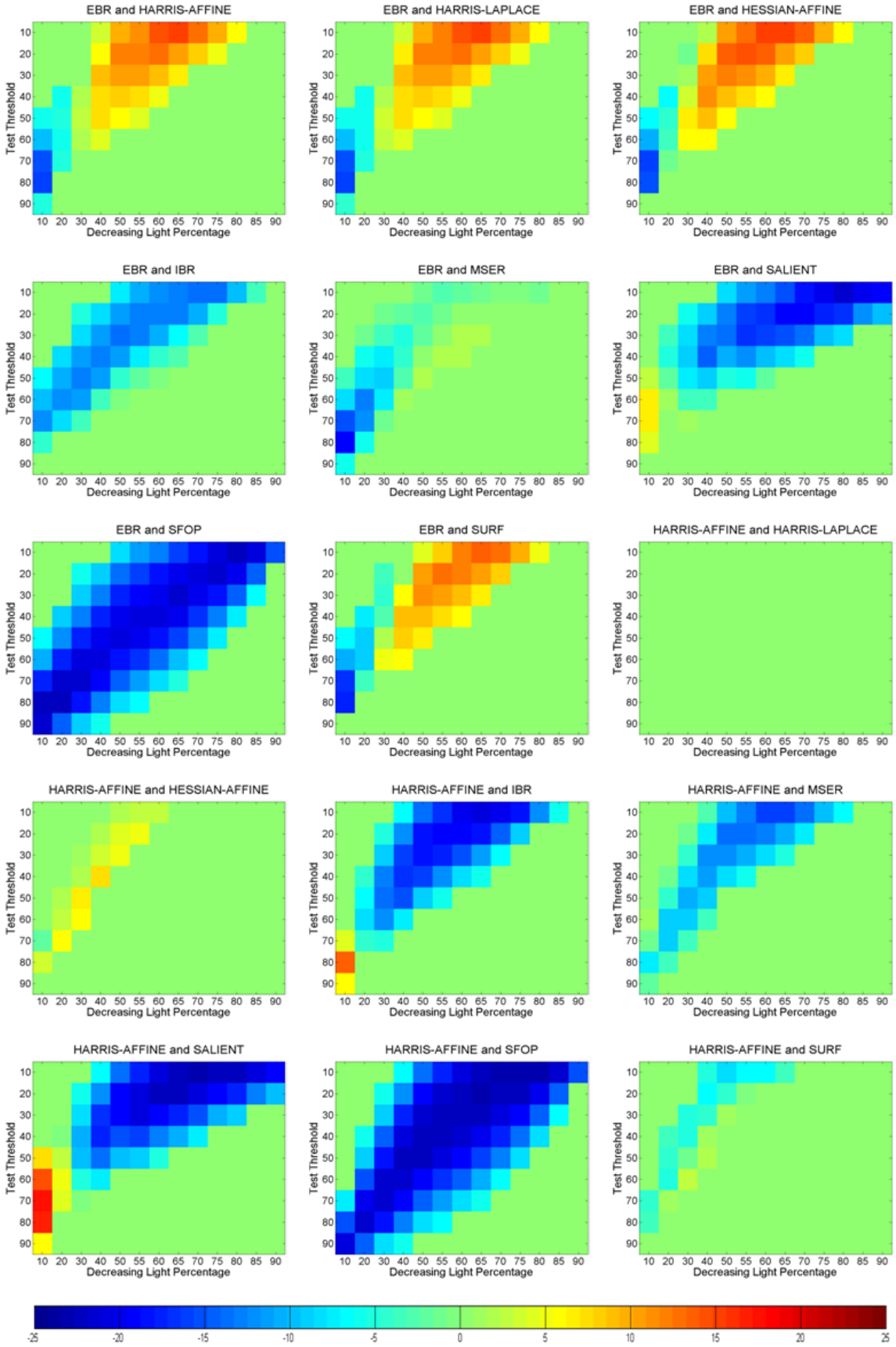

Figure 4-44: Light database results for EBR and Harris-Affine with the other detectors showing Z-scores obtained using the proposed framework; positive Z-scores indicate that the first detector is better than the second whereas the negative values show the converse

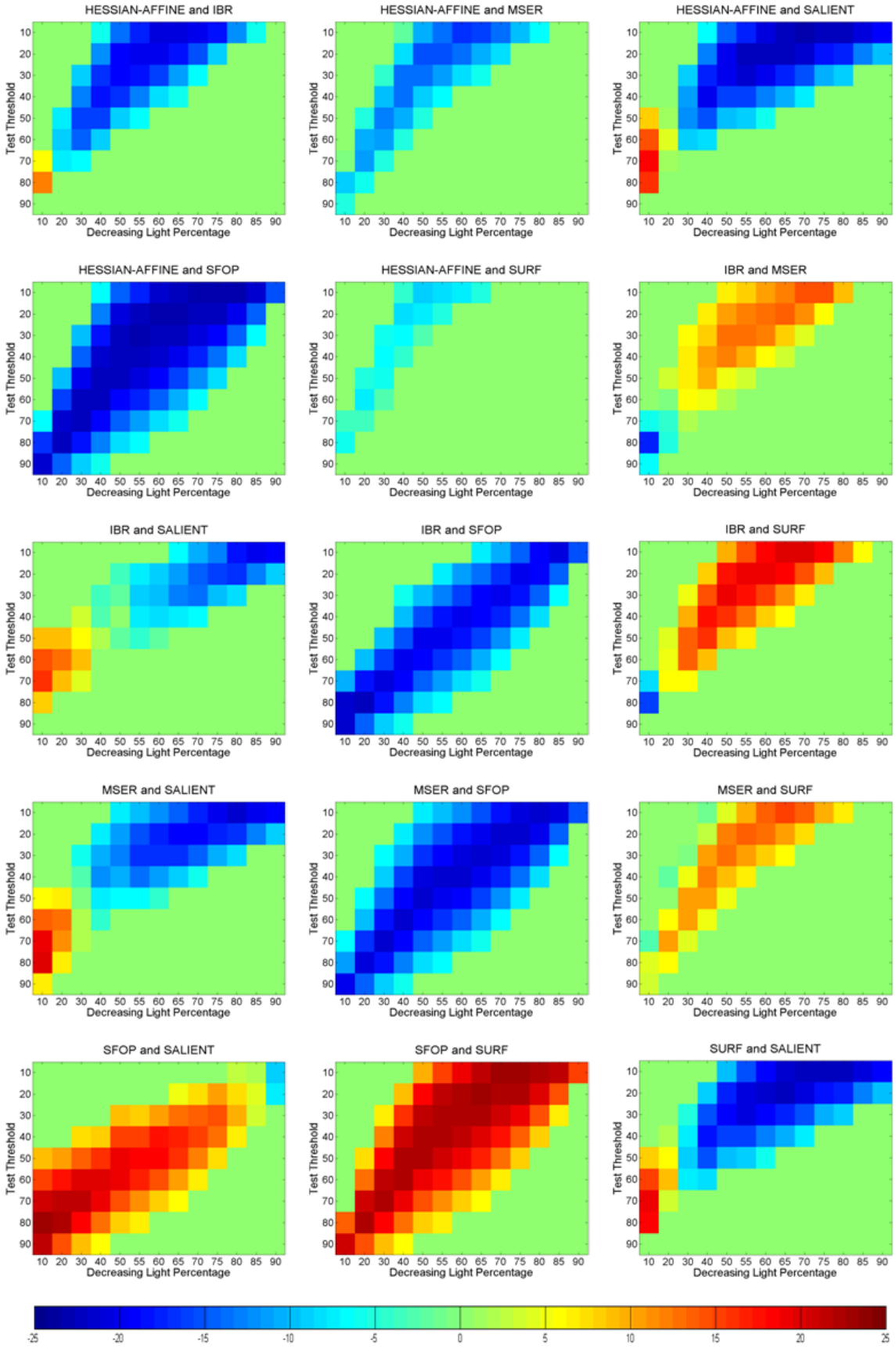

Figure 4-45: Light database results for Hessian-Affine, IBR, MSER, SFOP and SURF with other detectors showing Z-scores obtained using proposed framework; positive Z-scores show that the first detector is better than the second whereas negative values show the converse

4.6 Proposed Solution for Uniform Light Changes

As shown in the previous section, the state-of-the-art local feature detectors undergo a rapid decline in performance for even a small uniform change in light conditions, which is not desirable from a vision systems design viewpoint. This section presents a solution for improving the performance of these feature detection techniques under uniform changes in illumination. For demonstrating the effectiveness of the proposed solution, results for several state-of-the-art feature detectors utilizing the light image database (see Section 4.5.1) are presented which show significant improvement in performance.

4.6.1 Method

The method essentially adds a simple pre-processing stage to any feature detection technique in an effort to counter the negative effects of uniform changes in light on its performance. The method comprises of three distinct steps:

1) The arithmetic mean μ_o and standard deviation σ_o of the input image are calculated.

2) The pixel values of the input image are adjusted to achieve a target arithmetic mean μ_n and a target standard deviation σ_n by using the following equation:

$$P_{new} = \left(\frac{(P_o - \mu_o)}{\sigma_o} \times \sigma_n\right) + \mu_n \qquad \textit{Equation 4-9}$$

where P_o is an original pixel value and P_{new} is the adjusted one. As most distributions are symmetric, $2^{no.of\ bits\ of\ pixel-1}$ is selected as the value for μ_n to be used in Equation 4-9. For an 8-bit image pixel, the value of μ_n will be 128. The value of σ_n is chosen to be $\left(2^{no.of\ bits\ of\ pixel}\right)/6$ since 99.5% of Gaussian distributed values lie

within ±3 standard deviations of the arithmetic mean. For an 8-bit image pixel, the value of σ_n is 256/6.

3) The adjusted pixel values obtained from Equation 4-9 are clipped in the range 0 to $2^{no.of\ bits\ of\ pixel} - 1$. This essentially means that all adjusted pixel values which are less than zero are made equal to the minimum possible value in the range whereas the values greater than $2^{no.of\ bits\ of\ pixel} - 1$ are made equal to the maximum possible value in the range. For an 8-bit image pixel, this range is 0 to 255.

4.6.2 Results for State-of-the-art Detectors

The results for several state-of-the-art feature detectors using the proposed solution are presented in Figure 4-46 to Figure 4-54 for the light image database. It is evident that there is a marked improvement in the performances of detectors as compared to the results presented in Section 4.5 for uniform changes in light. Hessian-Laplace, Harris-Laplace, Hessian-Affine, Harris-Affine, SFOP and SURF show stable behavior for decreasing light conditions as indicated by their narrow *operating regions*. The *guarantee regions* of these detectors are also wide enough to show that there is minor degradation in the performances of detectors with decreasing light. MSER and IBR also show significant improvement as compared to the results presented in Figure 4-33 and Figure 4-34. However, barring EBR, their performance is overshadowed by the other detectors. Of all the detectors, EBR performs the worst but it still achieves better repeatability scores when compared to the results presented in Figure 4-36. Even for 90% decrease in light, all the above-mentioned detectors including EBR show reasonable performance, which provides evidence of the effectiveness of the proposed solution.

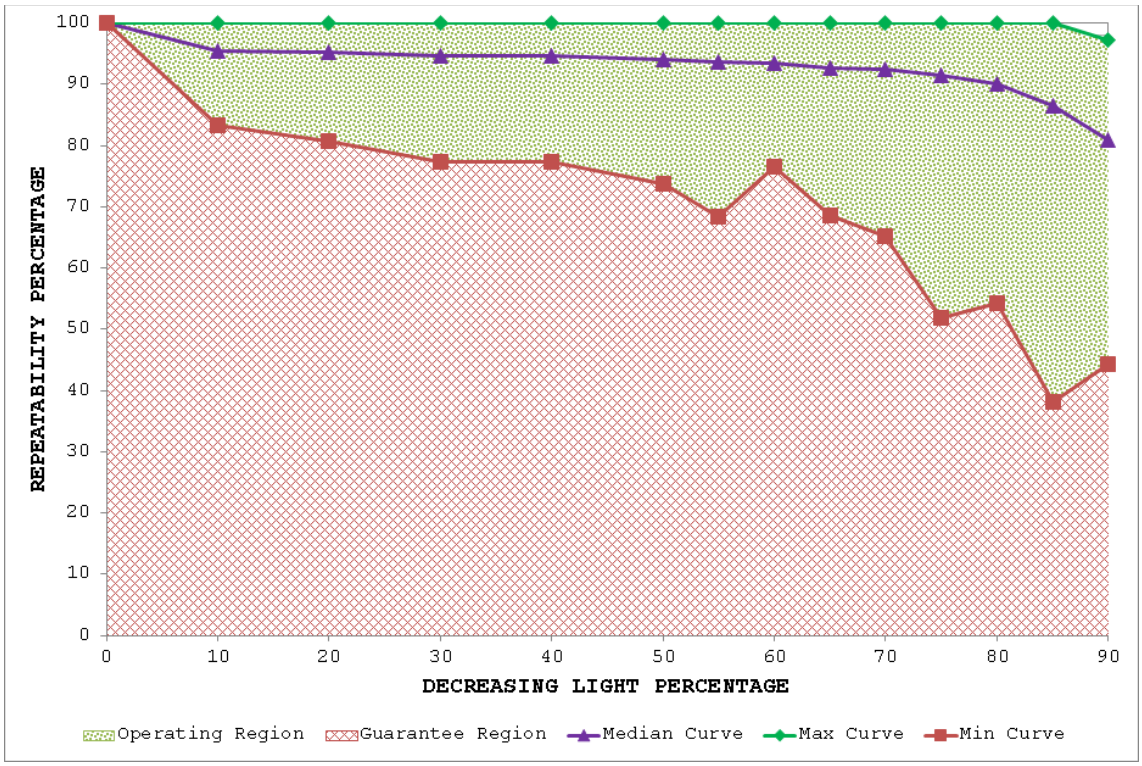

Figure 4-46: Light database results for MSER with the proposed method

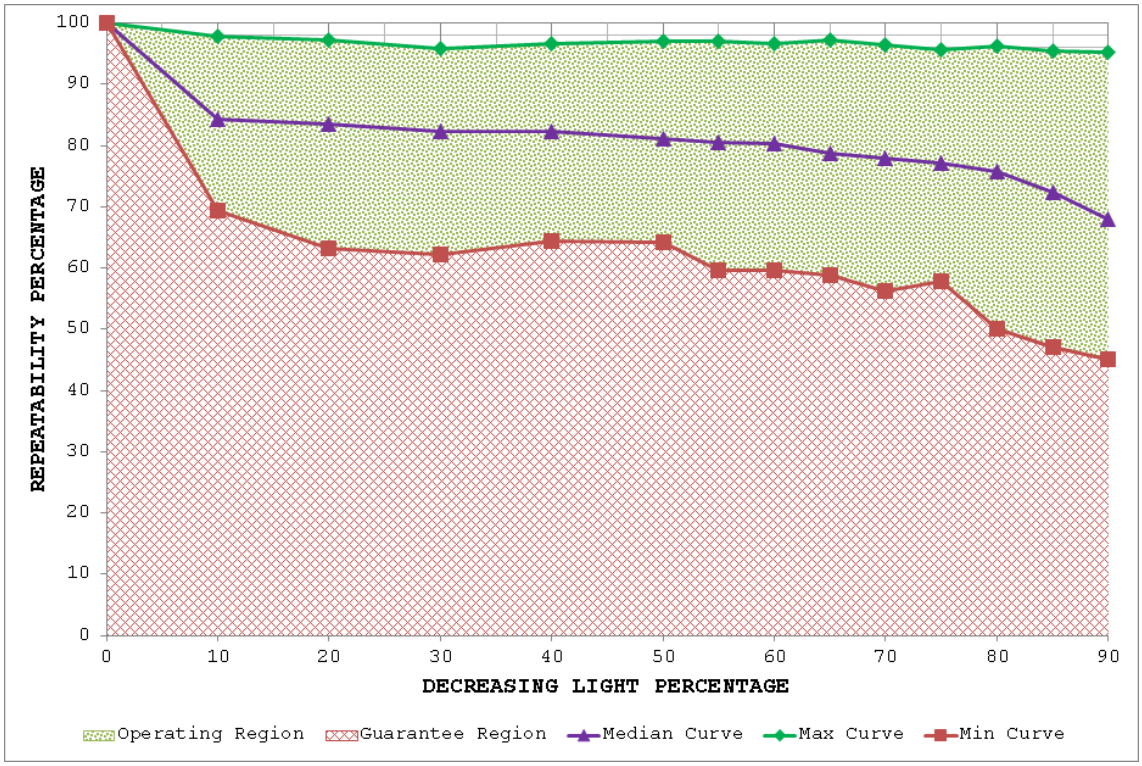

Figure 4-47: Light database results for IBR with the proposed method

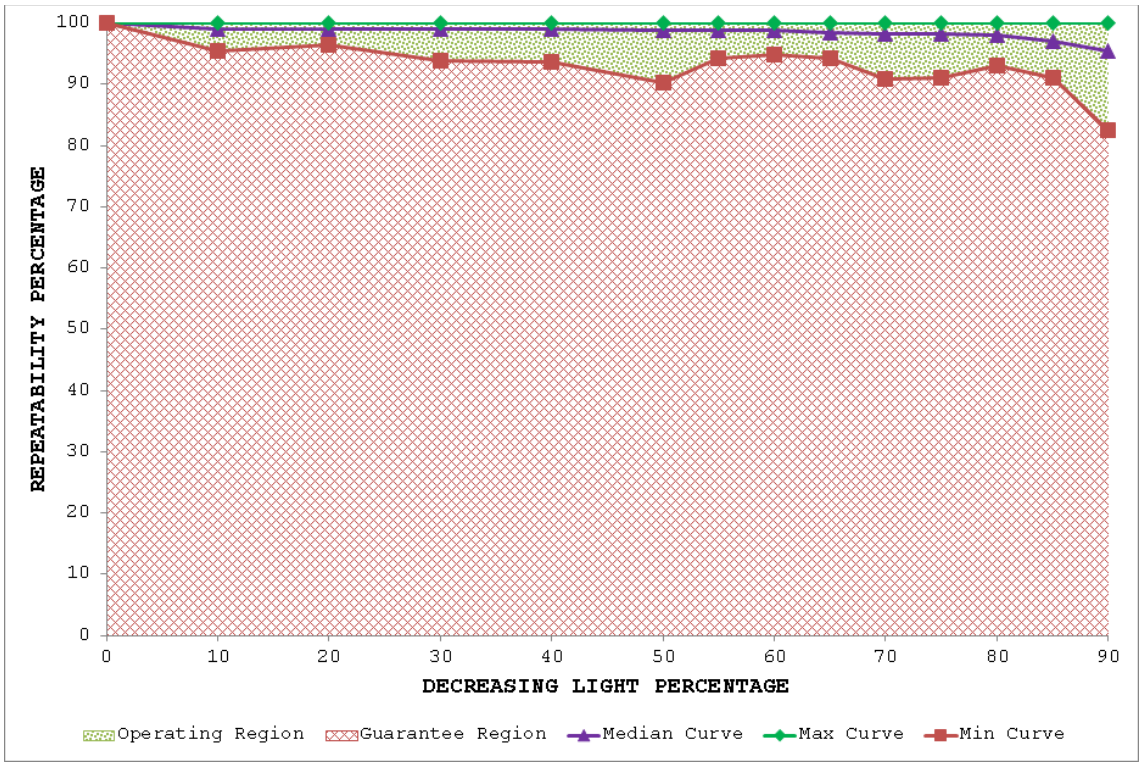

Figure 4-48: Light database results for Hessian-Laplace with the proposed method

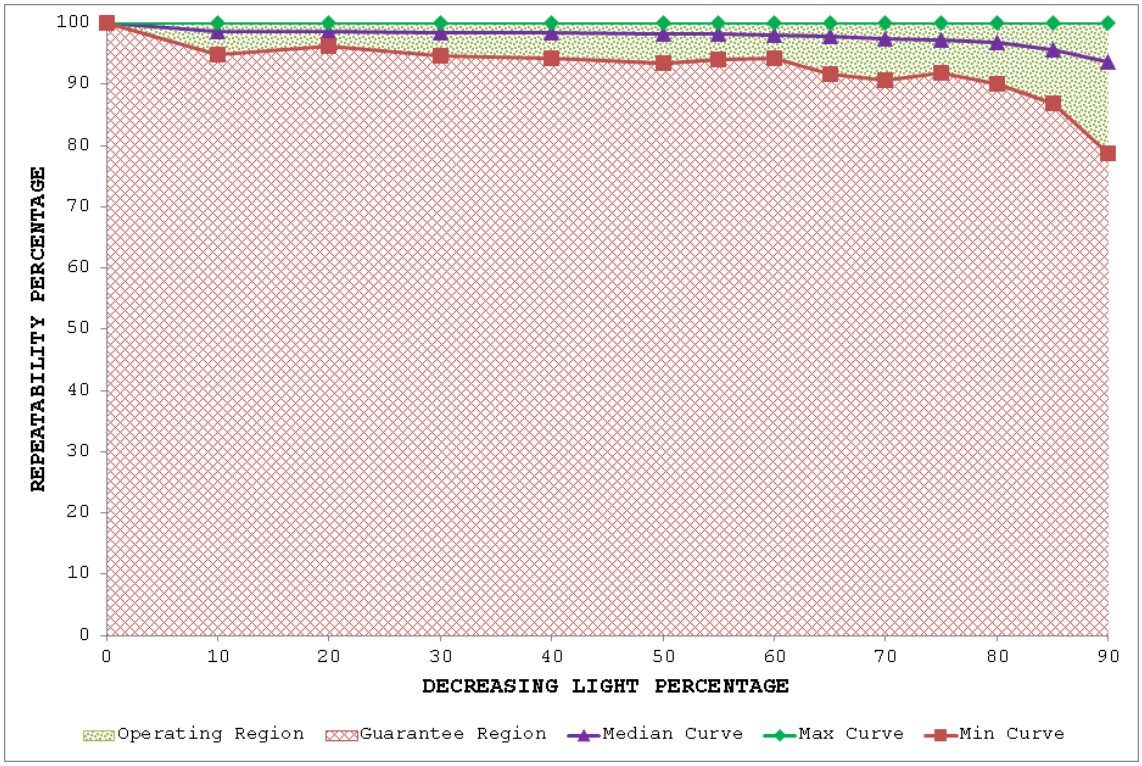

Figure 4-49: Light database results for Harris-Laplace with the proposed method

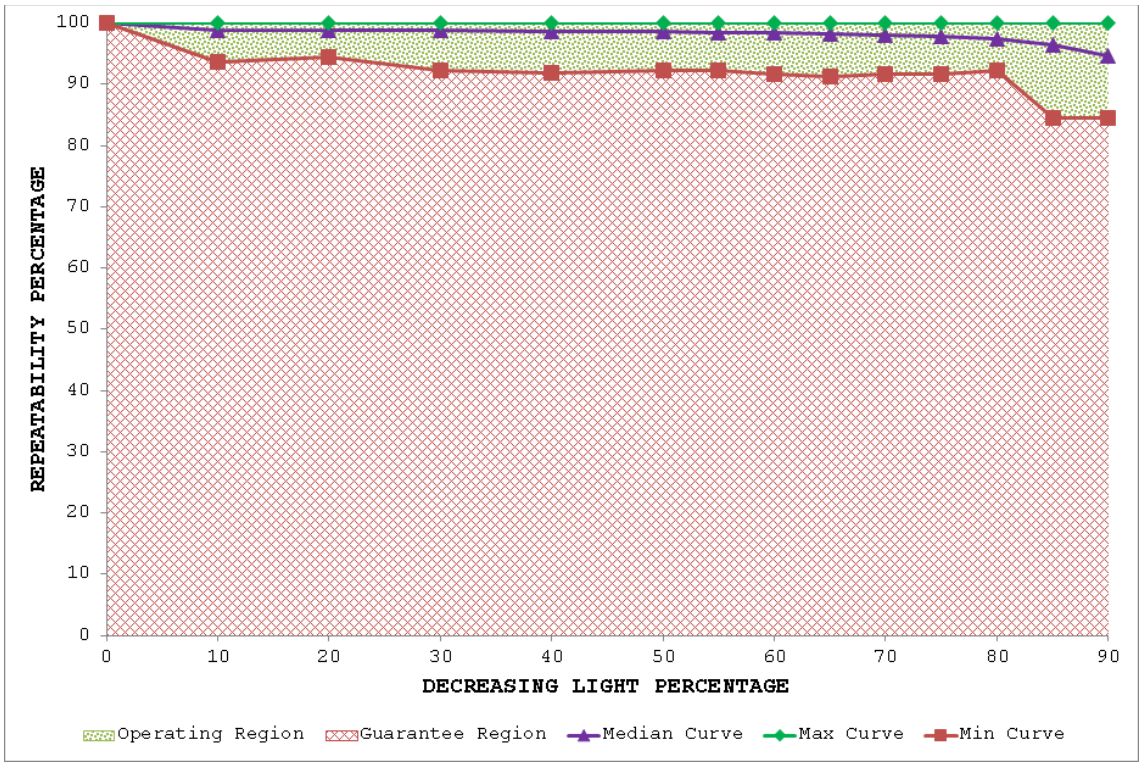

Figure 4-50: Light database results for SURF with the proposed method

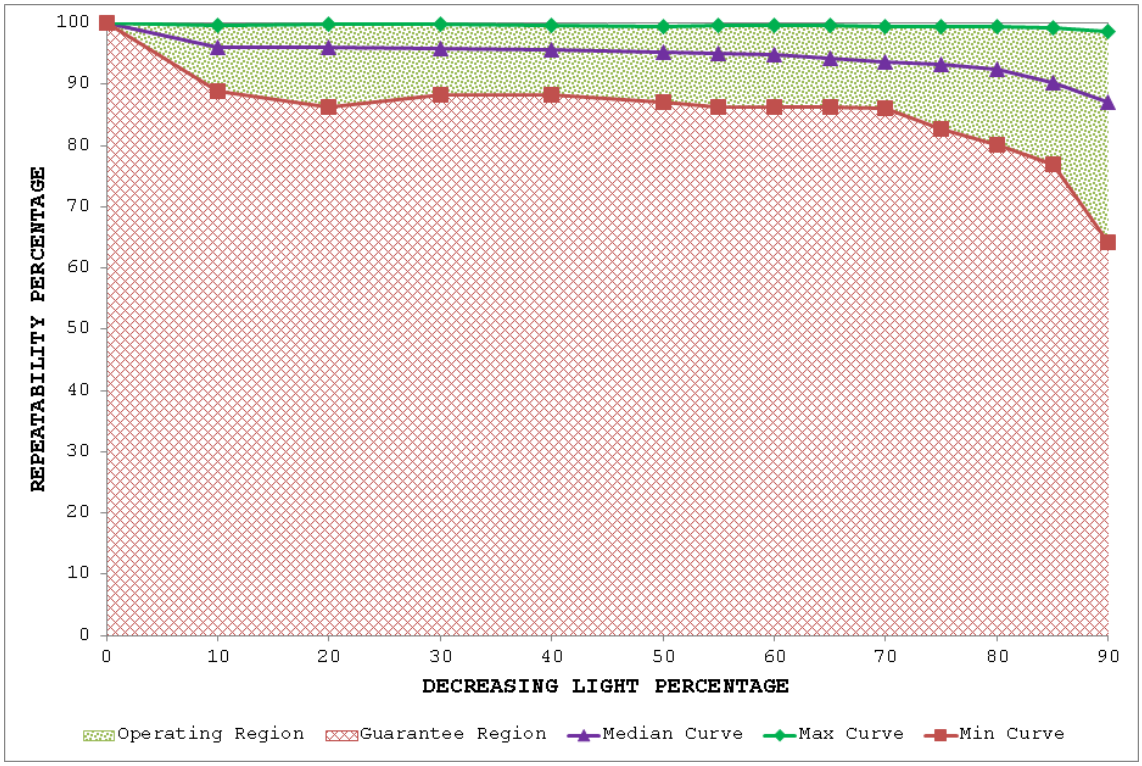

Figure 4-51: Light database results for SFOP with the proposed method

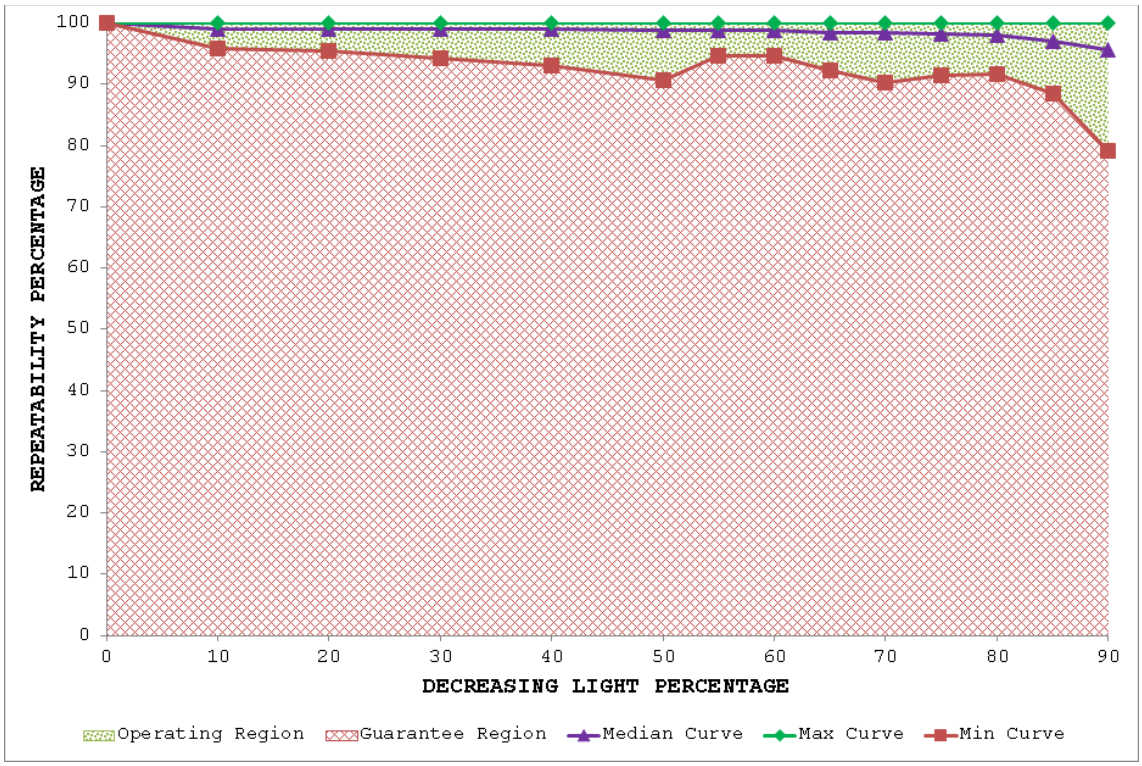

Figure 4-52: Light database results for Hessian-Affine with the proposed method

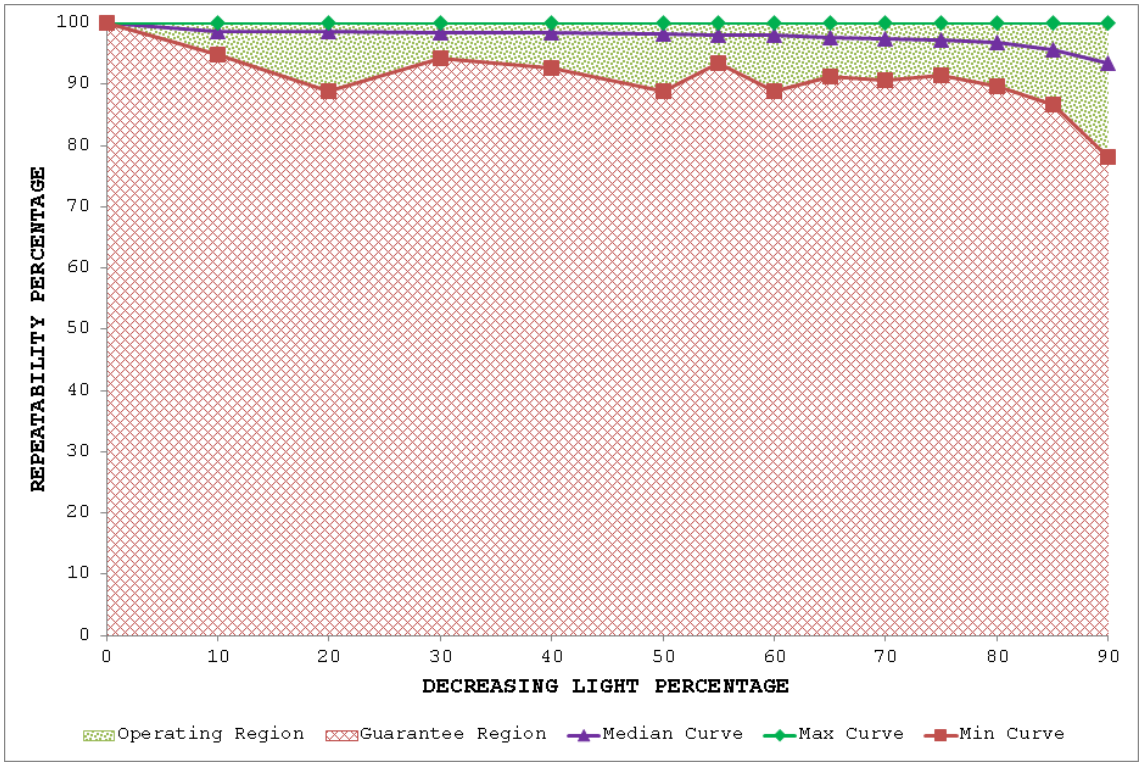

Figure 4-53: Light database results for Harris-Affine with the proposed method

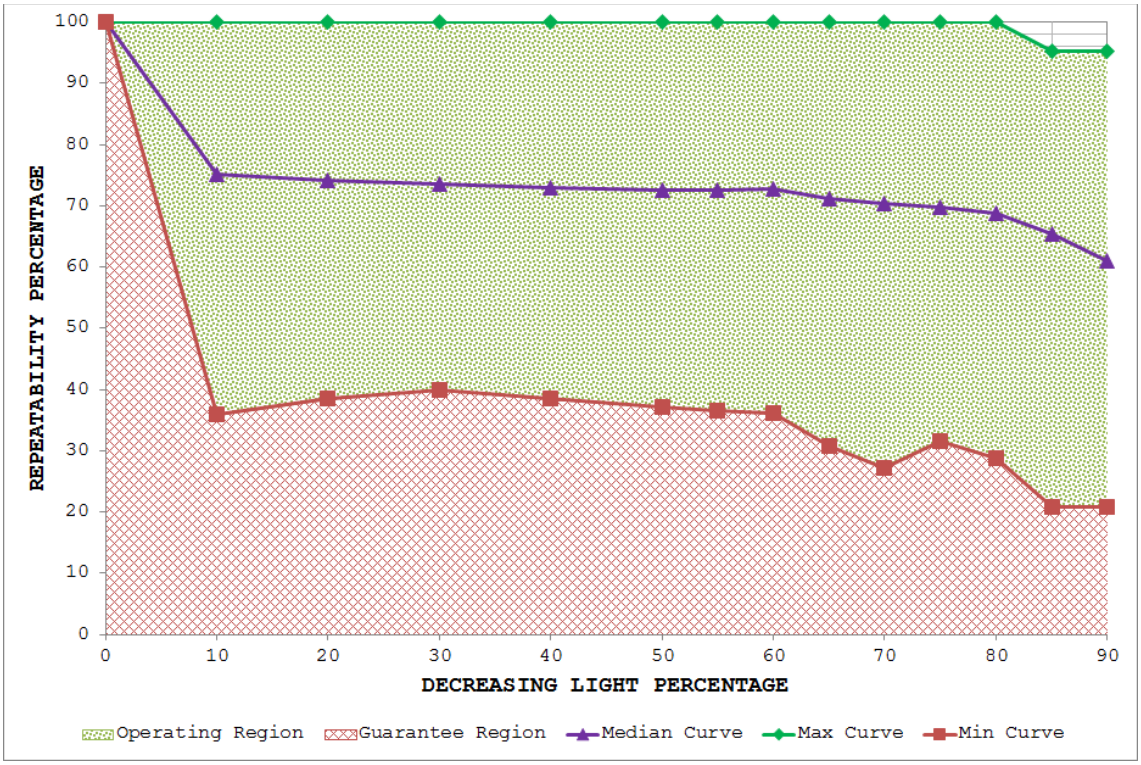

Figure 4-54: Light database results for EBR with the proposed method

4.7 Summary

For designing reliable and more effective vision systems, this chapter has presented a generic framework based on one of the improved repeatability measures proposed in Section 3.3.3. The framework has two important steps: the first one is the estimation of the upper and lower bounds of performance for a given feature detector under a specific image transformation in order to segment detector performance into *operating* and *guarantee* regions; and the second step is the identification of statistically significant performance differences between detectors as a function of the amount of image transformation. To that end, the chapter has introduced a variant of McNemar's test to find statistically significant performance differences. It has demonstrated the utility of the proposed framework by establishing *operating* and *guarantee* regions for several state-of-the art detectors and has identified statistical performance differences between them under JPEG compression, uniform light changes and blurring. Results are obtained by utilizing newly acquired, large image databases for JPEG

compression (7546 images with 539 different scenes), blur (5390 images with 539 different scenes) and uniform illumination changes (7546 images with 539 different scenes). These detailed results provide novel insights into the strengths and weaknesses of the detectors from a vision system design perspective. Finally, the chapter has proposed the inclusion of a pre-processing step as part of any feature detection technique to improve performance under uniform illumination changes. Results are presented which show that this pre-processing step drastically improves performance for several state-of-the-art detectors in the presence of uniform light changes.

5 Rapid Online Analysis of Local Feature Detectors and their Complementarity

Measure what is measurable, and make measurable what is not so.

GALILEO GALILEI

A vision system that can assess its own performance and take appropriate actions online to maximize its effectiveness would be a step towards achieving the long-cherished goal of imitating humans. This chapter proposes a method for performing an online performance analysis of local feature detectors, the primary stage of many practical vision systems. It advocates the spatial distribution of local image features as a good performance indicator and presents a metric that can be calculated rapidly, concurs with human visual assessments and is complementary to existing offline measures such as repeatability. The metric is shown to provide a measure of complementarity for combinations of detectors, correctly reflecting the underlying principles of individual detectors. Qualitative results on well-established datasets for several state-of-the-art detectors are presented based on the proposed measure. Using a hypothesis testing approach and a newly-acquired, larger image database, statistically-significant performance differences are identified. Different detector pairs and triplets are examined quantitatively and the results provide a useful guideline for combining detectors in applications that require a reasonable spatial distribution of image features, such as image registration and accurate multi-view geometry estimation. A principled framework for combining feature detectors in these applications is also presented. Timing results reveal the potential of the metric for determining the performance of detectors and their complementarity in online applications.

5.1 Introduction

The last decade has seen significant interest in the development of low-level vision techniques that are able to detect, describe and match image features [1, 12-14, 16, 18]. The most popular of these algorithms operate in a way that makes them reasonably independent of geometric and photometric changes between the images being matched. Indubitably, the Scale Invariant Feature Transform (SIFT) [12] has been the operator of choice since its inception and has provided the impetus for the development of other techniques such as Speeded-Up Robust Features (SURF) [13] and the Scale Invariant Feature Operator (SFOP) [16].

One of the main driving factors in this area is the improvement of detector performance. In Chapters 3 and 4, the author focused on repeatability [15, 46], the ability of a detector to identify the same image features in a sequence of images, which is considered a key indicator of detector performance and is the most frequently-employed measure in the literature for evaluating the performance of feature detectors [1]. However, it has been emphasized that repeatability is not the only characteristic that guarantees performance in a particular vision application [1, 184]; attributes such as efficiency and the density of detected features are also important. It is therefore desirable to be able to characterize the performance of a feature detector in several complementary ways, rather than relying only on repeatability [1, 129, 185]. Moreover, it is not possible to compute repeatability online in practical applications as doing so involves 'ground truth' data which are generally not available. Hence, a performance measure that can be calculated rapidly to assess detector performance online would be useful.

One property that is crucial for the success of any feature detector is the spatial distribution of detected features, known as the coverage [129]. Many applications, such as tracking and narrow-baseline stereo, require a reasonably even distribution of detected interest points across an image to yield accurate results; however, it is sometimes found that the features

identified by detectors are concentrated on a prominent textured object and hence cover only a small region of the image. Robustness to occlusion, accurate multi-view geometry estimation, accurate scene interpretation and better performance on blurred images are some of the advantages of detectors whose features cover images well [129, 185].

Despite its significance, there is no standard metric for measuring the coverage of feature detectors [129]. An approach based on the convex hull is employed in [45] to measure the spatial distribution of detected features. However, the convex hull traces the boundary of interest points without considering their density within that boundary and, as will be demonstrated in Section 5.2, results in an over-estimation of coverage. The convex hull approach is criticized by [19] and an alternative measure, completeness, presented. Completeness, however, employs an entropy coding scheme and Gaussian image model; results may vary with other coding schemes and image models, so this approach merits further investigation. Moreover, the metric is compute-intensive and so cannot be employed online for evaluating performance.

To fill this void, this chapter explores the online analysis of local feature detectors, proposing a metric that can be computed rapidly to measure the spatial distribution of detected features. It is intended to be used only with detectors that are known to have similar performances with offline measures such as repeatability and robustness to geometric and photometric transformation; this eliminates the possibility of favoring a poor detector that randomly scatters its points everywhere in the image. It can also be utilized in a framework such as [185] which is dependent upon the coverage of interest points, including those that cannot be matched accurately. Unlike repeatability [46, 47, 186], which is essentially a theoretical measure due to its requirement for ground truth, the proposed measure is a viable performance indicator for detectors in practical applications that require a reasonable distribution of detected features (assuming similar performances with offline measures). It will be demonstrated that the proposed measure concurs with human visual

assessments and is reliable. By employing a statistical hypothesis testing approach, a quantitative evaluation based on the proposed measure will be carried out to ascertain the statistical significance of performance differences between several state-of-the-art local feature detectors.

Since the notion of complementary feature detectors (i.e., combinations of detectors that identify different types of feature) was introduced by [187], they have become more popular for vision tasks [30, 188, 189]. Hence, it is valuable to have a measure of the complementarity of combinations of feature detectors so that their combined performance can be predicted and measured [1]. This chapter shows how mutual coverage, the coverage of a combination of the interest points from multiple detectors, can be used to measure complementarity and presents results from empirical investigations for combinations of detectors that reflect their underlying principles. The chapter also highlights the potential of the proposed measure as an online analysis tool for complementarity—the first of its kind, to the author's knowledge.

The remainder of the chapter is structured as follows: Section 5.2 describes the coverage measure, which is used to evaluate the performances of the eleven state-of-the-art feature detectors on well-established datasets encountered in Chapter 3. In order to avoid inadvertent data dependencies, Section 5.3 presents results obtained by employing statistical hypothesis testing on a new database of 520 images using the proposed coverage measure for the same detectors. A complementarity measure derived from coverage, termed mutual coverage, is proposed in Section 5.4 and its effectiveness is demonstrated by results for combinations of detectors. Section 5.5 discusses the feasibility of the proposed measures for real-world scenarios and demonstrates their speed advantage from a computational perspective. A framework for combining feature detectors in applications which require reasonable distribution of feature points is proposed in Section 5.6. Finally, a summary of the chapter is presented in Section 5.7.

5.2 Measuring Coverage

This section presents a method for measuring the spatial distribution of detector responses rapidly that makes it suitable for use in practical applications. Qualitative results on the widely-used Oxford datasets [50] are presented for the eleven state-of-the-art feature detectors to demonstrate the effectiveness of the measure.

5.2.1 Proposed Method

There are several *desiderata* for a coverage measure:

1) *Consistency with human visual inspection.* Humans can easily distinguish between a set of features that cover only a small region and one that is well-distributed over the whole image. The differences in spatial distribution of two sets of features indicated by the measure should be consistent with those obtained by human visual inspection.

2) *Penalization of clustered feature sets.* As stated in Section 5.1, it is quite common for local feature detectors to detect many feature points near a prominent textured object in an image. A useful measure would penalize techniques that concentrate interest points in a small region as that does not improve coverage.

3) *Avoidance of over-estimation.* The measure should avoid over-estimation of coverage by taking into account the density of feature points. To illustrate this, consider the simple example in Figure 5-1. Assuming that the four points shown in the image on the left are the output of a local feature detector for an image of size 640 x 480, the region enclosed by the dotted line is the convex hull of these four points. The ratio of the area of the convex hull to the area of the image, as used in [45], shows that these four points cover nearly 32% of the area of the entire image. If an additional interest point is detected inside the same region, as shown in the right-hand image of

Figure 5-1, the coverage reported by this measure is unchanged, despite there being an improvement in the spatial distribution of points. This is certainly not desirable.

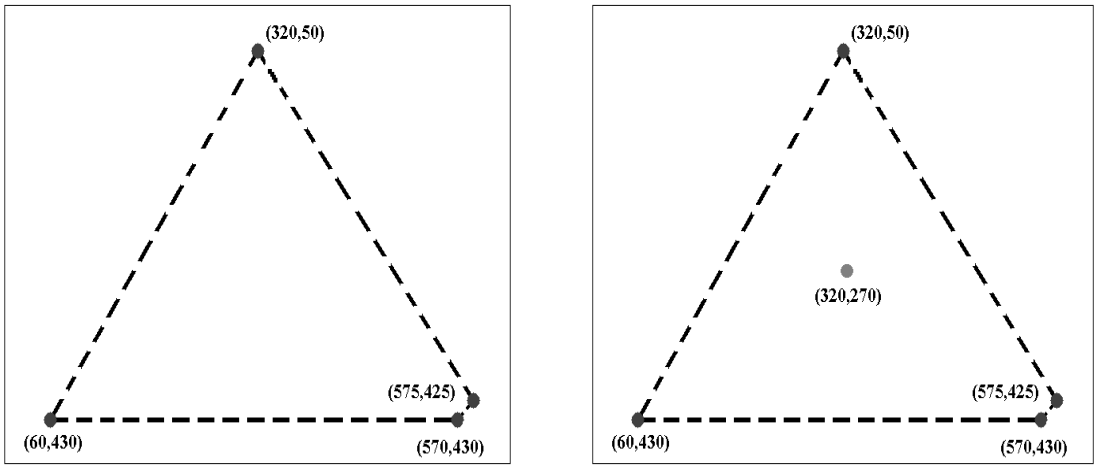

Figure 5-1: A simple example: (left) an image with four detected interest points and their convex hull; (right) the same image with an additional detected interest point and convex hull

4) *Homogeneous or non-textured regions*. Most local feature detectors work with high-entropy areas in the image. Consequently, homogeneous or non-textured regions have long been considered uninteresting by the vision community. However, the development of methods like NF-features [190] has shown the utility of non-textured regions in feature detection and matching. Unlike [19], which penalizes features appearing in homogeneous areas, the author argues that a good coverage measure should encompass all repeatable features, irrespective of the texture of the region in which they are detected.

5) *Ground truth information*. As mentioned above, repeatability, the most-widely employed performance measure for feature detectors, relies on the availability of ground truth, ultimately limiting its use to offline evaluation only. The completeness measure proposed in [19] requires calculation of entropy density of the entire image for use as reference, also making it unsuitable for online use. A metric that does not require ground truth information or reference computation would

be valuable for online applications. Since it is assumed here that all regions of the image are equally important for feature detection irrespective of the image content and texture (see point 4), it automatically eliminates the requirement to compute a reference.

6) *Low computation cost.* Online performance analysis of a feature detector can help it adapt to the nature of the imagery it is processing. However, existing performance measures for local feature detectors allow only offline evaluation due to their high computation cost. A measure that can be computed quickly is therefore required to achieve the goal of online performance analysis.

The obvious way to estimate coverage is to calculate the arithmetic mean of the Euclidean distance between feature points. However, the arithmetic mean is greatly influenced by outliers and may provide misleading estimates, especially for skewed distributions. The geometric mean also estimates the central tendency of a sample space in a way that is influenced by outliers, although less so than the arithmetic mean. Conversely, large outliers have little effect on the harmonic mean while small values are much more significant, making it good at penalizing clustered features while being reasonably robust to noise. These properties have led to its widespread use in data clustering algorithms [191]. Indeed, the harmonic mean is an inherently conservative approach for estimating the central tendency of a sample space, as

$$A(x_1, \ldots, x_N) \geq G(x_1, \ldots, x_N) \geq H(x_1, \ldots, x_N) \qquad \textit{Equation 5-1}$$

where $A(.)$ is the arithmetic, $G(.)$ the geometric and $H(.)$ the harmonic mean of the sample set $x_1,\ldots, x_N$, $x_i \geq 0\ \forall i$.

Formally, we assume that $p_1,\ldots,p_N$ are the N interest points detected by a feature detector in image $I(x,y)$, where x and y are the spatial coordinates. Taking p_i as a reference interest point, the Euclidean distance d_{ij} between p_i and some other interest point p_j is

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad \textit{Equation 5-2}$$

providing $i \neq j$. Computation of Equation 5-2 provides $N-1$ Euclidean distances for each reference interest point p_i. The harmonic mean of d_{ij} is then calculated to obtain a mean distance D_i, $i = 1, \ldots, N$ with p_i as reference:

$$D_i = \frac{N-1}{\sum_{j=1, j\neq i}^{N} \left(\frac{1}{d_{ij}}\right)} \qquad \textit{Equation 5-3}$$

Since the choice of the reference interest point can affect the calculated Euclidean distance, this process is repeated using each interest point as reference in turn, resulting in a set of distances D_i. Finally, the coverage of the feature detector is calculated as

$$Coverage = \frac{N}{\sum_{i=1}^{N} \left(\frac{1}{D_i}\right)} \qquad \textit{Equation 5-4}$$

Since multi-scale feature detectors may provide image features at exactly the same image location but different scales, interest points that result in zero Euclidean distance in Equation 5-2 are excluded from the calculations on the basis that they do not improve the spatial distribution of features. It is clear from Equation 5-4 that coverage has the dimension of length (*i.e.*, pixels), so its value needs to be considered against the image dimensions as the same coverage value may indicate a good distribution for a small image but a poor distribution for a large one, a topic that is considered in more detail in Section 5.3.3. In general, a large coverage value is desirable for a feature detector as a small value implies the concentration of interest points into a small region.

To illustrate the advantage of the proposed measure over the convex hull approach [45], the simple example of Figure 5-1 is utilized again. For the case of four detected points (the image on the left), the proposed coverage measure provides a small value (39.49) to reflect that, although there are some widely-spaced points, the density of points is low. The

coverage value for the case that includes the additional interest point in the right-hand image of Figure 5-1 is 50.26, indicating an improvement in the spatial distribution of feature points.

5.2.2 Qualitative Results

For the proposed coverage measure to have any value, its values need to be consistent with visual assessments of coverage across a range of feature detectors and a variety of images. To that end, this section presents a comparison of the coverage of the eleven state-of-the-art feature detectors encountered in Chapter 3: SIFT (Difference-of-Gaussians), SURF (Fast Hessian), Harris-Laplace, Hessian-Laplace, Harris-Affine, Hessian-Affine, Edge-based Regions (EBR), Intensity-based Regions (IBR), Salient Regions, Maximally Stable Extremal Regions (MSER) and Scale Invariant Feature Operator (SFOP) [1, 16]. These were chosen because they are representative of a number of different approaches to feature detection (see Section 5.4.2 and [1]); also their implementations are widely available and they have broadly similar repeatability performance. Although the control parameters of these feature detectors can be varied to yield a similar number of interest points for all detectors, this approach has a negative effect on their repeatability and performance [19]. Therefore, authors' original programs (binary or source) have been utilized with parameters set to values recommended by them, and the results presented were obtained with the widely-used Oxford datasets [50]. The parameter settings and the datasets used make these results a direct complement to existing evaluations.

To demonstrate the effectiveness of this coverage measure, first consider the case of the Leuven dataset [50] in Figure 5-2. It is evident that SFOP outperforms the other detectors in terms of coverage, whereas values for EBR, Harris-Laplace and Harris-Affine indicate a poor spatial distribution of interest points. To back up these results, the actual distribution of detector responses for SFOP, IBR, Harris-Laplace and EBR for image 1 of the Leuven dataset are presented in Figure 5-3. Visual inspection of these distributions is consistent with the coverage results of

Figure 5-2: the interest points detected by SFOP are distributed all over the image rather than being concentrated on a specific textured object in Figure 5-3. IBR also seems to achieve a reasonable spatial distribution of interest points. On the other hand, the image features detected by EBR and Harris-Laplace appear clustered in small regions and fail to cover the image well, a fact that is correctly reflected by Equation 5-4 (see Figure 5-2).

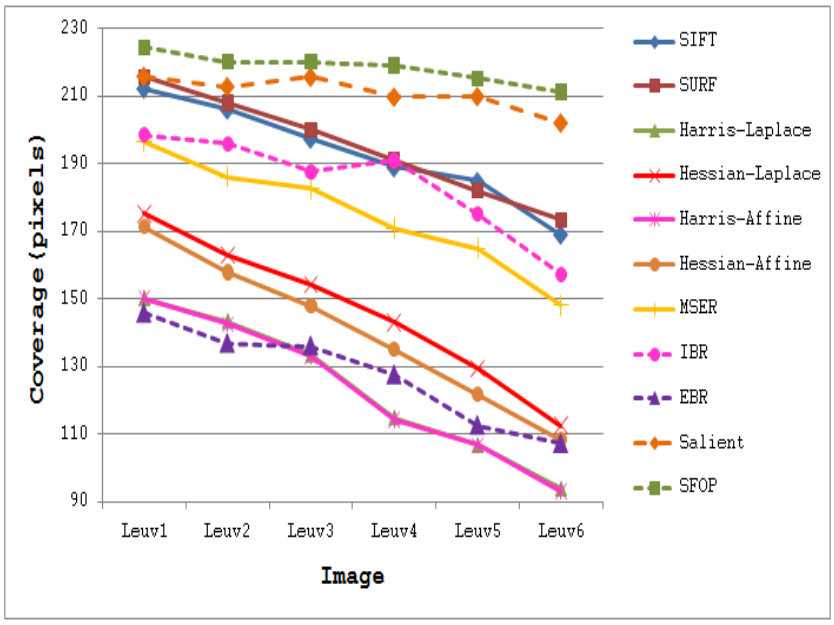

Figure 5-2: Coverage results for Leuven dataset [50]

The coverage values obtained for the Boat dataset [50] are presented in Figure 5-4. Again, the performances of well-established techniques like SIFT and SURF are eclipsed by SFOP. Harris-Laplace, Harris-Affine, Hessian-Affine and EBR again fare poorly. In addition, the curves depicted in Figure 5-2 and Figure 5-4 incorporate the effects of illumination changes (Leuven dataset) and zoom and rotation (Boat dataset) on coverage.

A summary of the mean results obtained with all these feature detectors for the Oxford datasets [50] is presented in Table 5-1. It is clear that SFOP achieves better coverage than the other feature detectors for almost all datasets under various geometric and photometric transformations.

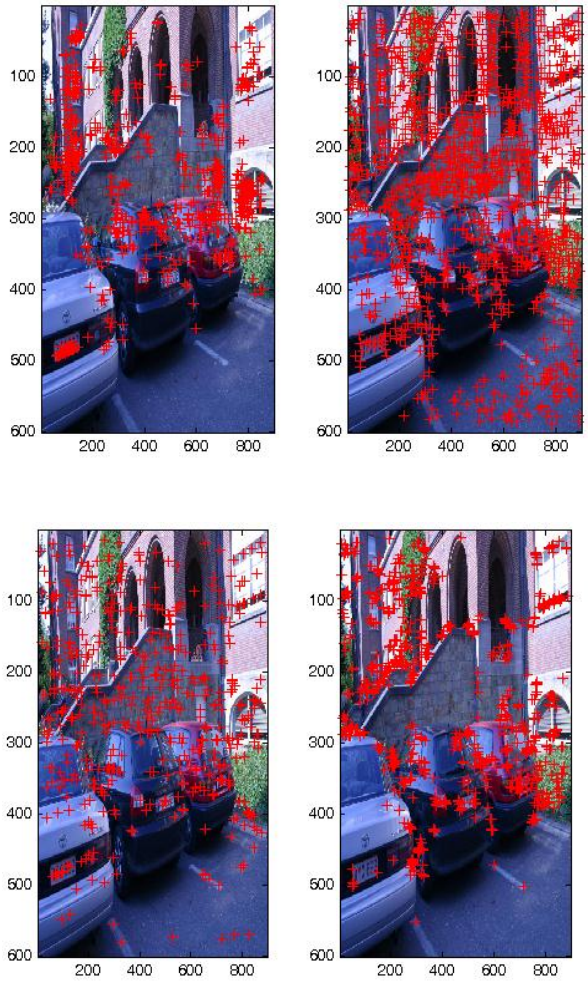

Figure 5-3: Actual detector responses for image 1 of Leuven dataset [50]. From top left to top right: EBR and SFOP; from bottom left to bottom right: IBR and Harris-Laplace

Table 5-1: Coverage results for state-of-the-art feature detectors

	Bark	Bikes	Boat	Graffiti	Leuven	Trees	UBC	Wall
SIFT(DoG)	190.3	207.8	206.7	221.0	193.1	263.4	204.2	253.5
SURF(FH)	195.8	228.1	207.8	221.9	195.0	265.4	205.4	246.6
Harris-Lap	122.9	136.5	143.6	181.2	123.7	230.2	154.5	213.7
Hessian-Lap	120.0	154.5	154.8	199.2	146.2	234.2	154.9	208.6
Harris-Aff	122.8	136.0	142.8	181.0	123.3	229.9	153.8	212.8
Hessian-Aff	119.9	148.9	146.5	191.0	140.4	233.0	153.5	208.2
Salient	190.6	258.7	213.8	218.0	211.0	256.4	201.5	236.4
EBR	139.2	138.3	119.1	166.4	127.7	214.3	119.0	204.4
IBR	192.3	214.7	189.7	209.7	184.2	255.5	198.4	243.8
MSER	179.6	86.4	177.0	200.3	174.9	229.6	200.6	248.3
SFOP	204.4	246.3	224.4	228.7	218.3	270.3	213.8	256.5

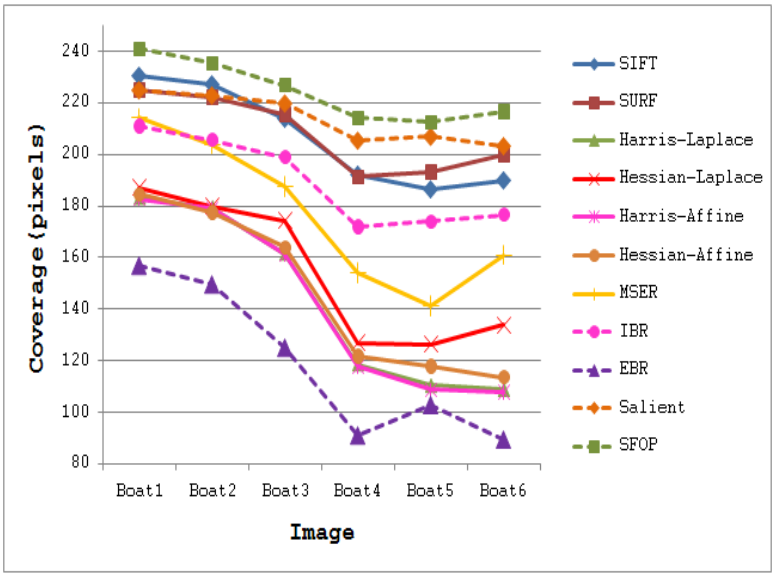

Figure 5-4: Coverage results for the Boat dataset [50]

5.3 Performance Evaluation

Although the results presented in Section 5.2 on the widely-used Oxford datasets complement existing evaluations, the small number of images involved makes drawing statistically-significant conclusions difficult. Hence, a confirmatory data analysis is required to ascertain whether or not the obtained results have occurred by chance due to inadvertent data dependencies, and to do this a larger database of images needs to be used. The confirmatory data analysis revolves around two important questions:

1) Do the results obtained for the Oxford datasets provide a complete insight into the behavior of feature detectors? In other words, are the results obtained for the Oxford datasets consistent with the results obtained on a larger image database, having a variety of scenes and variations in texture?

2) Are differences in coverage between various feature detectors statistically significant?

A discussion of the methodology employed to tackle the above questions and the results obtained are given below. A third important

question, asking whether high coverage implies good performance in an application, is considered in Section 5.5.

5.3.1 The Image Database

With the objective of yielding statistically-valid comparisons of coverage-based performance, the author has captured a database of 520 images, more than ten times the size of the Oxford datasets. Since the distribution of detected local features is dependent upon the nature of the imagery, such as natural scenes and man-made objects, it is quite possible that a specific type of content may favor a particular detector during performance analysis. This issue has been addressed by including images with a variety of scene types, categorized into four datasets based on content: Snow, Indoor, Campus-1 and Campus-2. This categorization allows identification of the strengths and the weaknesses of detectors with regards to image content. Each dataset contains more than 100 images of 1440 x 956 pixels, with structured and non-structured scenes and medium to low levels of texture. For example, the Snow dataset includes images that have large areas of scene covered with snow, leading to low texture. Similarly, most images in the Indoor dataset contain one or two prominent objects in low-texture surroundings. Some images from these four datasets are shown in Figure 5-5. To facilitate comparisons of other feature detectors with the author's findings, these image datasets are made available at [192].

5.3.2 Quantitative Evaluation on Image Database

To answer the first question, coverage values for the eleven state-of-the-art detectors of Section 5.2 were calculated using the large image database [192], again utilizing binaries provided by the authors and the recommended parameter settings. Since every detector included in this evaluation generally extracts different numbers of interest points for a given image, the mean number of features detected by each detector for the four image datasets is depicted in Figure 5-6 so as to determine its possible impact on coverage. It is clear that SIFT, SURF and Salient detect large

numbers of interest points for all datasets, whereas the feature sets extracted by other detectors are relatively sparse. The mean coverage results obtained with all these feature detectors for the Snow, Indoor, Campus-1 and Campus-2 datasets [192] are shown in Figure 5-7 to Figure 5-10 respectively. It should be noted that, following [19], the error bars in these figures indicate the 1-σ confidence intervals for the mean values, where σ is the probability of Type I error. The associated confidence level with these intervals is 95%, which is often used in practice [193].

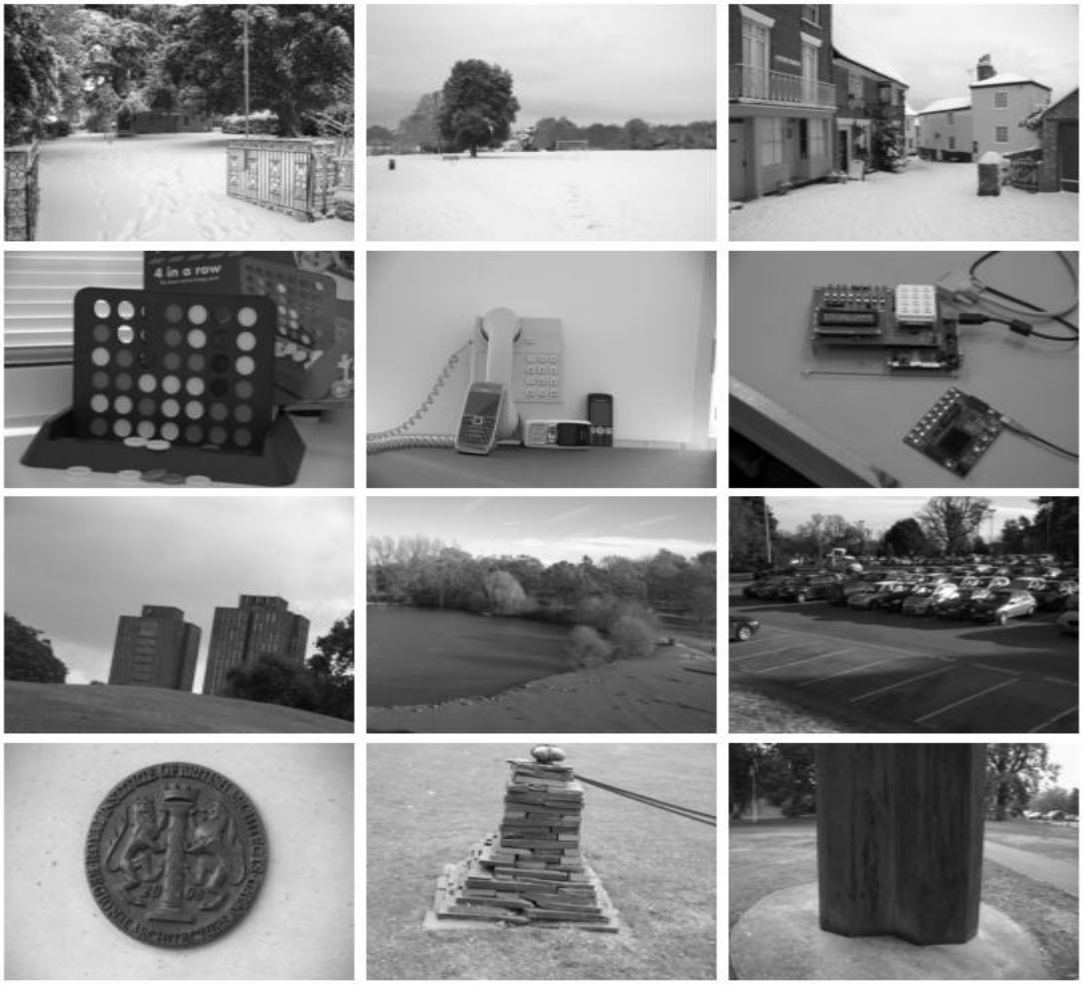

Figure 5-5: Some images from the Snow, Indoor, Campus-1 and Campus-2 datasets in the first, second, third and fourth row respectively

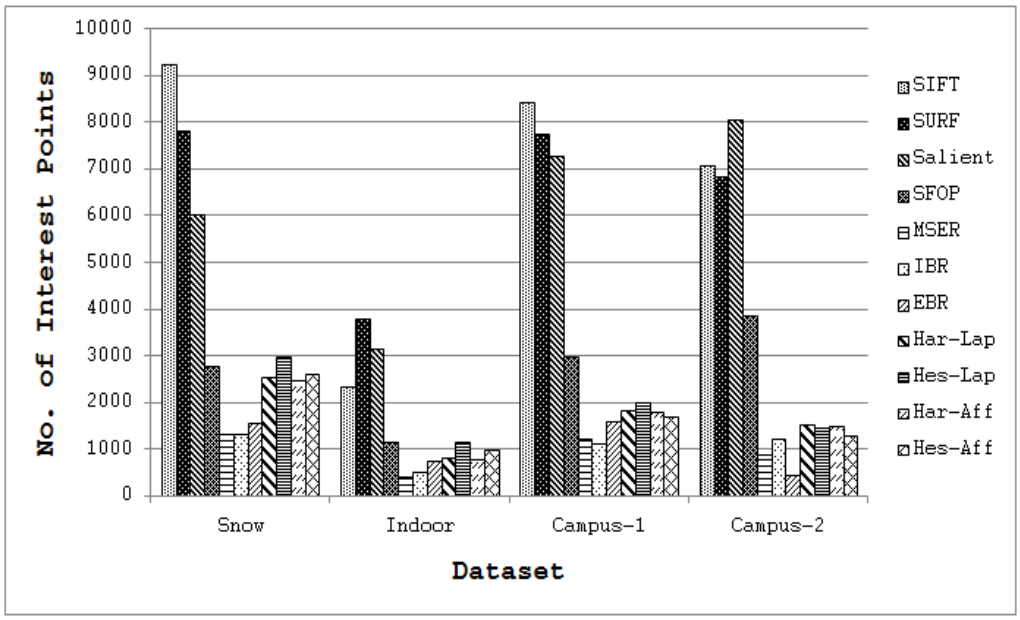

Figure 5-6: Average number of interest points detected by state-of-the-art detectors on image database [192]

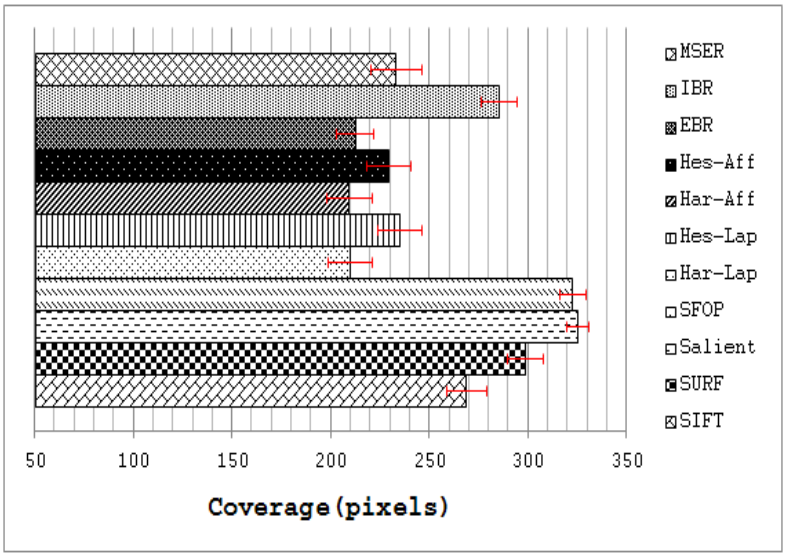

Figure 5-7: Coverage results for Snow dataset [192]; the error bars indicate the 95% confidence intervals for mean values

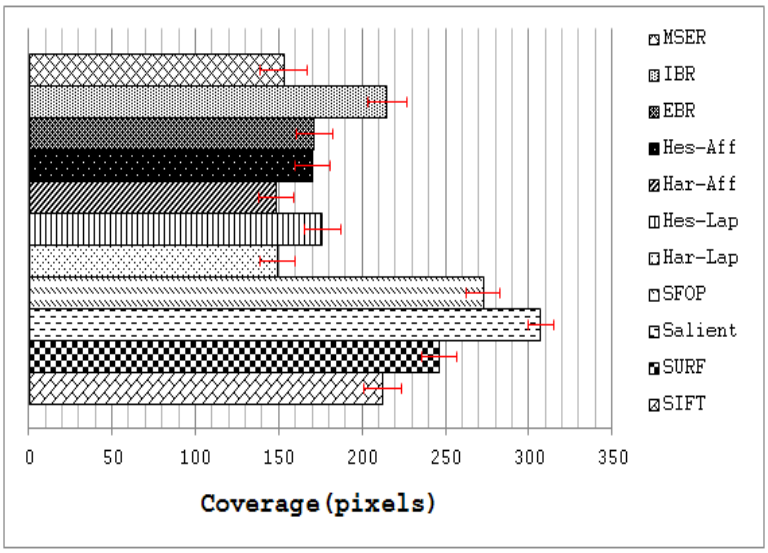

Figure 5-8: Coverage results for Indoor dataset [192]; the error bars indicate the 95% confidence intervals for mean values

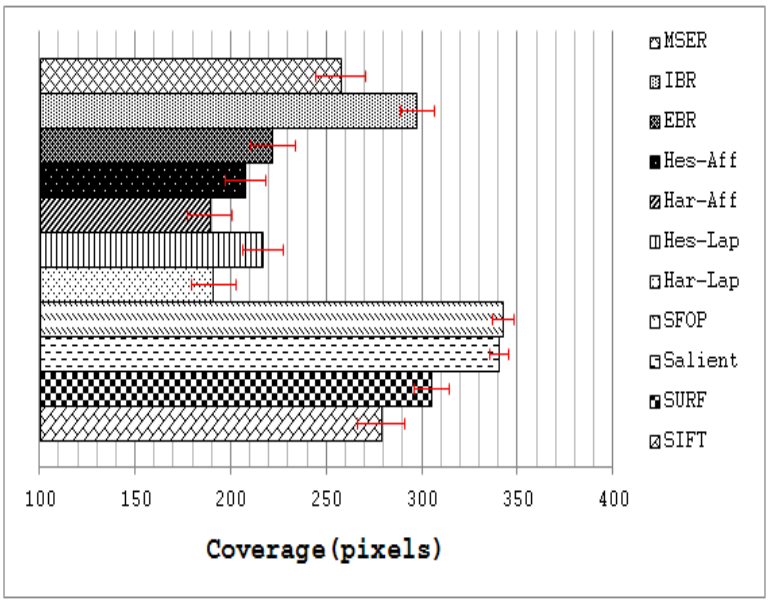

Figure 5-9: Coverage results for Campus-1 dataset [192]; the error bars indicate the 95% confidence intervals for mean values

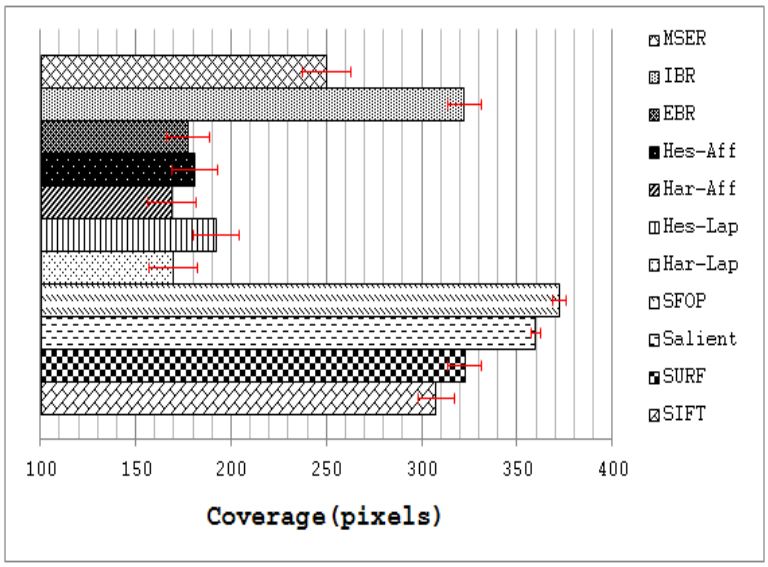

Figure 5-10: Coverage results for Campus-2 dataset [192]; the error bars indicate the 95% confidence intervals for mean values

Although the results obtained on the image database appear broadly consistent with the findings for the Oxford datasets, there are some discrepancies. It is evident from Figure 5-7 to Figure 5-10 that SFOP and Salient provide the best coverage. Apart from the Indoor and Campus-2 datasets, there is only a marginal difference between the mean coverage values achieved by SFOP and Salient for the other two image datasets. SFOP prevails in the case of Campus-2 but is out-performed by Salient for Indoor, a significant discrepancy from the results obtained for the Oxford datasets [50]—this can perhaps be attributed to the lack of indoor scenes in the Oxford datasets. On the other hand, the performance of SFOP can be considered remarkable considering that it generally detects fewer interest points than Salient. For example, for the first image of the Campus-1 dataset, Salient detects 8799 interest points whereas SFOP detects only 3348 points, roughly 2.5 times fewer. However, SFOP still achieves a better coverage value of 333.1 as compared to Salient (326.44).

Figure 5-7 to Figure 5-10 show that SURF out-performs SIFT in terms of coverage — again, a digression from the results obtained in Section 5.2. In addition, the performance of SIFT is eclipsed by IBR for all four datasets, which is not apparent in the results presented in Section 5.2. A

reasonable explanation for this might be the availability of a limited number of scenes with texture variations in the Oxford datasets. MSER achieves relatively good coverage values for the Campus-1 and Campus-2 datasets, both of which contain images with good to medium levels of texture, but its performance is poor for the more challenging Snow and Indoor datasets. Also, the Hessian-Laplace and Hessian-Affine detectors perform slightly better than their Harris-based counterparts. It is evident that EBR fails to achieve good coverage values for all four datasets.

5.3.3 Identifying Statistically-Significant Performance Differences

Since 1-σ confidence intervals for population means do not necessarily indicate statistically significant results [194, 195], it is desirable to perform some statistical tests that ascertain whether any differences in performances between different feature detection algorithms are statistically significant in order to back up the largely qualitative discussion of performance in Section 5.2. Formally, one proposes a null hypothesis (*i.e.,* that there is no difference in performance between methods) and uses a statistical test to determine whether the data are consistent with this hypothesis. Although statistical tests like ANOVA (analysis of variance), paired t-test and Wilcoxon signed rank test provide direct methods to assess the difference between population means depending upon distribution [193], the author finds it more useful to identify statistically-significant performance differences in a manner that can be related to the spatial distribution of interest points in the image. An appropriate statistic in this case is the non-parametric McNemar's test, a form of chi-squared test with one degree of freedom that evaluates the performance of the two algorithms based on their outcomes on a case-by-case basis over the same dataset [51, 52] (see Equation 4-6). The author has utilized McNemar's test to compare the performances of these eleven feature detectors for the large image database [192]. To employ it, one needs a criterion to determine whether a test case results in success or failure. As coverage has the dimension of a

length, a criterion that encapsulates the size of an image seems a suitable option for such an evaluation. A common such criterion in the physics literature that has long been used for specifying field sizes is the ratio of area to perimeter [196]:

$$Coverage \geq \frac{Area\ of\ Image}{Perimeter\ of\ Image} \qquad \textit{Equation 5-5}$$

More precisely, if an algorithm satisfies Equation 5-5, it is considered to have succeeded; otherwise, it is deemed to have failed. Although arbitrary, experiments show that this criterion is consistent with the visual inspections discussed in Section 5.2. For example, for the first image of the Leuven dataset [50], which has dimensions of 900 x 600 pixels, the area divided by perimeter is 180; detectors which satisfy Equation 5-5 exhibit good spatial distribution of interest points visually, whereas the others fare poorly (see Figure 5-2 and Figure 5-3).

An experiment was performed in which the coverage was calculated for each detector on every image in the database [192]. Where the coverage exceeded the threshold of Equation 5-5, the detector was deemed to have succeeded on that image; otherwise, it failed. This allowed N_{sf} etc (in Equation 4-6) for each pair of detectors to be determined over the image database and hence a *Z-score* calculated. Table 5-2 details the numbers of successes and failures for SFOP and Salient with the other detectors under consideration and the resulting *Z-scores*. Since it is not possible to include such detailed results for all detectors, a summary of the *Z-scores* for McNemar's tests between different detectors is given in Table 5-3, where positive values indicate that the detector in the left hand column performs better than the detector mentioned on the top and vice versa. Although the *Z-score* is always greater than or equal to zero, this sign convention is used to facilitate identifying the detector with the better performance of the two compared. *Z-scores* of about 3 are equivalent to a confidence of about 0.995, while larger *Z-score* values indicate a more significant result. It is clear that most values in Table 5-2 and Table 5-3 are substantially larger than 3 and

hence provide evidence that differences in coverage values between the detectors are statistically significant.

These results confirm the better performance of Salient and SFOP detectors over all other feature detectors considered. However, it is interesting to note that Salient out-performs SFOP, as there are 56 images for which SFOP failed to achieve good coverage but where Salient succeeded; conversely, there are only 10 images for which Salient failed and SFOP succeeded. The resulting Z for these results is 5.53, indicating that Salient detector out-performs SFOP with a probability well in excess of 0.995. Barring Salient, which detects two to three times more interest points (see Figure 5-6), SFOP appears to be the best detector of the remaining ones by a significant margin.

Apart from Salient and SFOP, high *Z-scores* were achieved by the SURF detector against all remaining detectors, including SIFT and IBR. Of the two segmentation-based detectors, IBR performs much better than MSER as indicated by a high *Z-score* of 11.96. The results also highlight that EBR ranks very low in terms of coverage-based performance. It is observed that Harris-Laplace and Harris-Affine behave in exactly the same manner ($Z = 0$) and fail to outperform EBR. Moreover, Hessian-Laplace barely manages to prevail over Hessian-Affine, as indicated by a low value of Z; this presumably reflects the similar underlying principles of the two detectors.

Table 5-2: McNemar's test results for SFOP and Salient detector with other detectors

	SIFT PASS	**SIFT FAIL**	**SURF PASS**	**SURF FAIL**	**SALIENT PASS**	**SALIENT FAIL**	**MSER PASS**	**MSER FAIL**
SFOP PASS	239	174	308	105	403	10	132	281
SFOP FAIL	1	106	1	106	56	51	1	106
Computed *Z*-Score	13.0		10.0		5.53		16.61	
	EBR PASS	**EBR FAIL**	**IBR PASS**	**IBR FAIL**	**HAR-LAP PASS**	**HAR-LAP FAIL**	**HES-LAP PASS**	**HES-LAP FAIL**
SFOP PASS	36	377	280	133	35	378	55	358
SFOP FAIL	1	106	0	107	0	107	1	106
Computed *Z*-Score	19.28		11.44		19.39		18.78	
	SIFT PASS	**SIFT FAIL**	**SURF PASS**	**SURF FAIL**	**MSER PASS**	**MSER FAIL**	**IBR FAIL**	**IBR FAIL**
SALIENT PASS	240	219	306	153	133	326	279	180
SALIENT FAIL	0	61	3	58	0	61	1	60
Computed *Z*-Score	14.73		11.92		18.0		13.23	
	EBR PASS	**EBR FAIL**	**HAR-LAP PASS**	**HAR-LAP FAIL**	**HES-LAP PASS**	**HES-LAP FAIL**	**HES-AFF PASS**	**HES-AFF FAIL**
SALIENT PASS	37	422	35	424	56	403	48	411
SALIENT FAIL	0	61	0	61	0	61	0	61
Computed *Z*-Score	20.49		20.54		20.02		20.22	

5.3.4 Discussion

It is valuable to correlate these performance differences to the underlying principles of the detectors in order to validate the proposed measure. Whilst responding to a number of different feature shapes, most feature detectors exhibit a strong response for a specific type of feature; for example, SIFT shows a bias for blobs in the image. Conversely, Salient is based on Shannon's entropy and responds equally to different feature types [18]; this allows it to achieve good coverage, though the large number of interest points detected also plays an important role in this regard. The design of SFOP utilizes several feature types in the same spirit as Salient, including star-like and circular shapes. The good ranking achieved by SFOP emphasizes the benefits of extracting multiple types of features.

Table 5-3: A summary of McNemar's test results (computed Z-score) for state-of-the-art detectors; negative values indicate that the detector mentioned on the top performs better than the detector shown on the left hand side

	SURF	MSER	IBR	EBR	HAR-LAP	HES-LAP	HAR-AFF	HES-AFF
SIFT	–6.90	10.15	–4.41	14.17	14.24	13.41	14.28	13.78
SURF	--	13.11	3.64	16.43	16.49	15.84	16.52	16.09
MSER	--	--	–11.96	8.89	9.42	7.56	9.47	8.19
IBR	--	--	--	15.39	15.58	14.76	15.62	15.03
EBR	--	--	--	--	0.17	–2.62	0.33	–1.52
HAR-LAP	--	--	--	--	--	–3.84	0	–2.50
HES-LAP	--	--	--	--	--	--	3.96	2.47
HAR-AFF	--	--	--	--	--	--	--	–2.77

As completeness and coverage serve similar purposes, it is also interesting to compare this ranking of detectors with the results presented in [19]. Salient is identified as the best detector in both studies. Although MSER is reported to have completeness scores comparable to those of Salient in [19], the rank for MSER here is lower than SFOP, IBR and SIFT. It is, however, agreed that the performance of MSER is commendable considering the sparseness of its features as compared to SFOP and SIFT.

In addition, the presented results suggest that SIFT is significantly better than the Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine detectors in terms of coverage. Since all these detectors, including SIFT, are stated to have similar completeness scores (see Fig. 12 in [19]), this observation is contradictory to [19].

5.4 Mutual Coverage for Measuring Complementarity

This section extends the coverage-based metric of Section 5.2 to measure the complementarity of combinations of detectors. After describing the mathematical formulation, the metric is utilized to present results for detector pairs and triplets.

5.4.1 Method

Since the utilization of combinations of detectors is an emerging trend in feature detection [1], the author proposes a measure, based on coverage, to estimate how well these detectors complement one another. In addition to the principles mentioned in Section 5.2, the objective here is to penalize techniques that detect several interest points in a small region of an image: if detectors A and B detect most feature points at the same locations in an image, they should have a low complementarity score. Conversely, a high score should be achieved if detectors A and B detect most features at widely-spaced locations, indicating that they complement each other well. Again, a metric utilizing the harmonic mean seems a promising solution to achieve the required goal, for the reasons discussed in Section 5.2.

Formally, let us consider an image $I(x, y)$, where x and y are the spatial coordinates, being operated on by M feature detectors $F_1, F_2, \ldots, F_M$, so that $P_z = \{P_{z1}, P_{z2} \ldots, P_{zN}\}$ is the set of N feature points detected by F_z. We then define

$$T_{zk} = P_z \cup P_k \qquad \textit{Equation 5-6}$$

as the set of feature points detected in image $I(x,y)$ by F_z and F_k. The coverage is then calculated as described in Section 5.2 using T_{zk}; as that includes points detected by both F_z and F_k, it is denoted as the *mutual coverage* of F_z and F_k for image $I(x,y)$. Although this chapter confines itself to combinations of two and three detectors only, this notion of mutual coverage can be extended to more by simply combining their feature points in Equation 5-6.

5.4.2 Results for Detector Pairs

To ascertain how well the detectors under discussion complement one another, the mutual coverages of combinations of these detectors were calculated. The author starts with the hypothesis that all detectors are complementary to one another and combines each detector with all other detectors in groups of two; if a pair's mutual coverage value is high, it should be because they identify different types of feature—in other words, a high mutual coverage should reflect their different principles of operation.

A categorization of the eleven feature detectors was published in [1] and is summarized in Table 5-4. This experiment allows us to ascertain whether or not this taxonomy requires revision to reflect the findings on the larger database employed here.

Table 5-4: A taxonomy of state-of-the-art feature detectors based on [1]

Category	Type	Detectors
1.	Blob detectors	SIFT, SURF, Hessian-Laplace, Hessian-Affine, Salient Regions
2.	Spiral detectors	Scale Invariant Feature Operator (SFOP)
3.	Corner detectors	Edge-based Regions (EBR), Harris-Laplace, Harris-Affine
4.	Segmentation-based detectors	MSER, Intensity-based Regions (IBR)

Figure 5-11 to Figure 5-17 depict the mean mutual coverages for the detectors under investigation when grouped with all other detectors for image database [192]. Note that the error bars in these figures indicate the 1-σ confidence intervals for mean values, with a confidence level of 95%. As

expected, all combinations involving Salient achieve good coverage (see Figure 5-11). The best results are obtained from a combination of Salient and SFOP, which is not surprising as both detect several types of features and have good individual coverages. Grouping Salient with IBR or MSER also provides good performance; this also reflects underlying principles, as the two segmentation-based detectors usually detect irregularly-shaped patterns and some blob-like structures, which helps to complement Salient. The combination of EBR and Salient also performs well, which again can be attributed to the different type of features they detect. Apart from Harris-Laplace and Harris-Affine, which start from the Harris corner detector, the detectors that yield low coverage values when combined with Salient (see Figure 5-11) are those that mainly detect blobs. A good explanation of this is the fact that Salient itself typically 'fires' on blob-like structures in the image. It is also interesting to note that SURF and SIFT perform the worst of all combinations involving Salient, despite detecting large number of interest points. (see Figure 5-6).

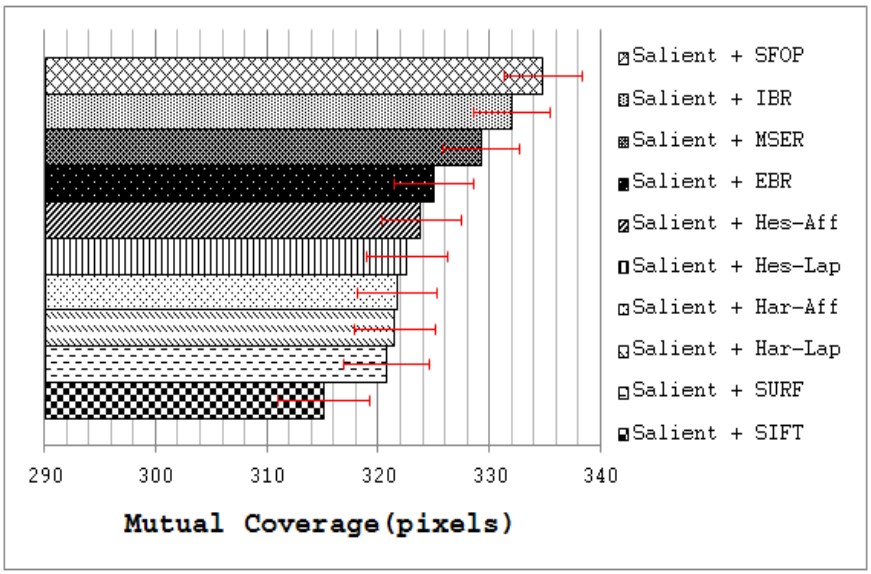

Figure 5-11: Mutual coverage of Salient detector in combination with other detectors for image database [192]; the error bars indicate the 95% confidence intervals for mean values

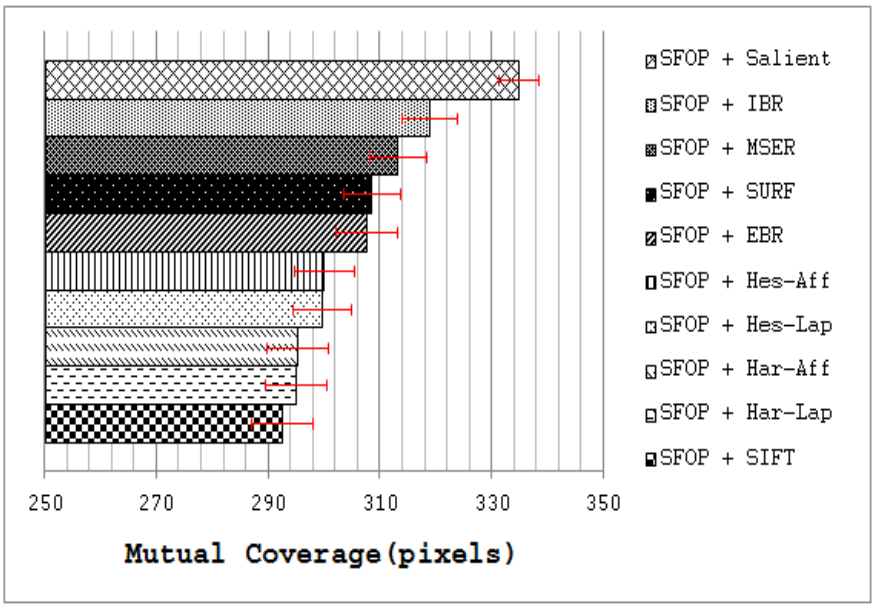

Figure 5-12: Mutual coverage of SFOP detector in combination with other detectors for image database [192]; the error bars indicate the 95% confidence intervals for mean values

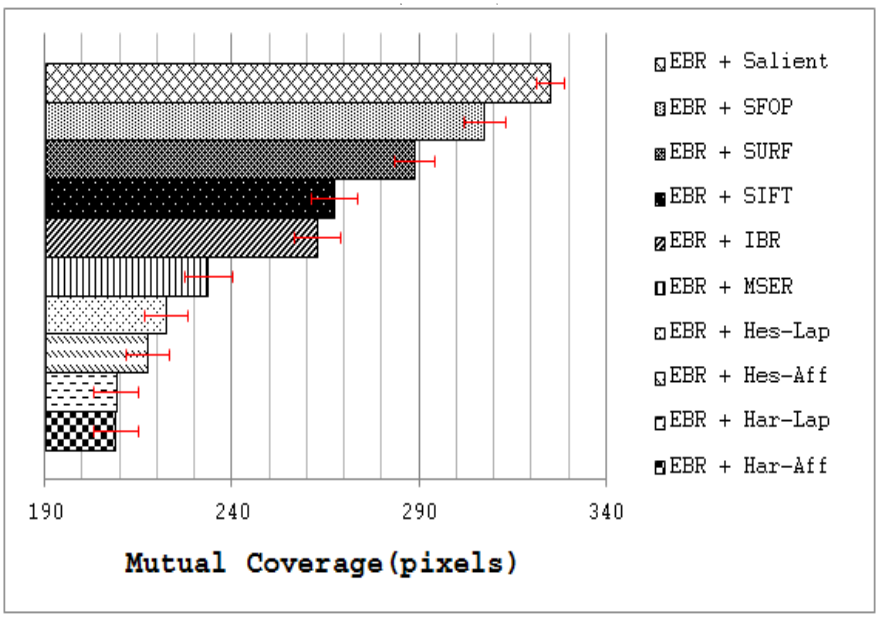

Figure 5-13: Mutual coverage of EBR in combination with other detectors for image database [192]; the error bars indicate the 95% confidence intervals for mean values

Apart from Salient, SFOP works best with IBR and MSER (as shown in Figure 5-12) which is again understandable due to the detection of different feature types. SURF and EBR also yield good coverage when combined with SFOP, for the same reason. Of all the remaining combinations involving SFOP, SIFT again performs worst, which may be attributed to the ability of SFOP to find some SIFT-like blobs in an image.

Figure 5-13 shows that combining SURF or SIFT with EBR achieves reasonable coverage. Grouping EBR with IBR or MSER is not particularly rewarding. Similarly, combinations involving Hessian-Laplace, Hessian-Affine, Harris-Laplace and Harris-Affine fare poorly.

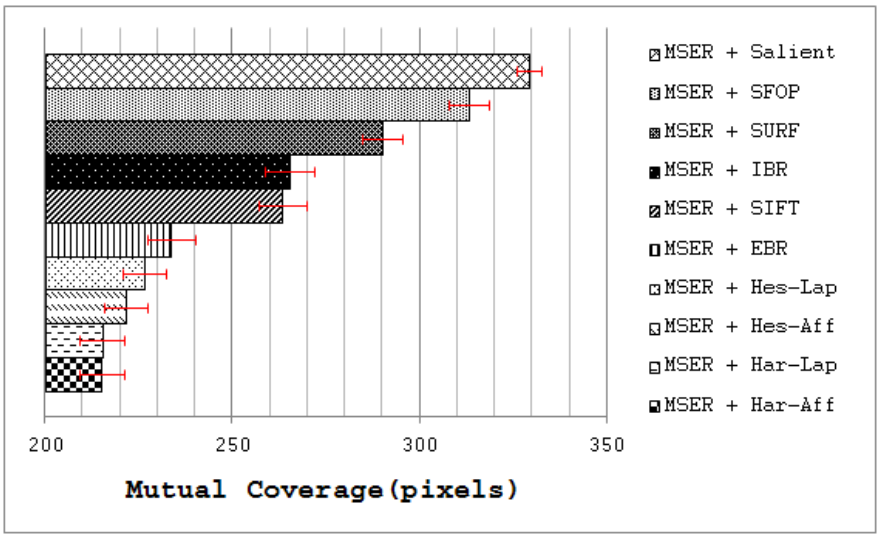

Figure 5-14: Mutual coverage of MSER in combination with other detectors for image database [192]; the error bars indicate the 95% confidence intervals for mean values

MSER and IBR often detect blob-like structures in an image in addition to irregularly-shaped patterns. Figure 5-14 and Figure 5-15 highlight that they work better with Salient and SFOP as compared to blob detectors. In Figure 5-14, it is interesting to note that a combination of MSER and IBR, which are somewhat similar in spirit, achieves higher coverage than a group involving MSER and SIFT. This shows that the feature sets of MSER and SIFT have some redundancy. On the other hand, IBR does not share this property and its combination with SIFT achieves higher coverage than a group of two segmentation-based detectors. Finally, it is evident from Figure 5-16 that combinations of SURF and SIFT with other blob detectors yield low coverage as compared to their combination with detectors that extract different feature type. Also, Hessian-Laplace, Hessian-Affine, Harris-Laplace and Harris-Affine, when combined with one another in a group of two, fare poorly as can be seen from Figure 5-17.

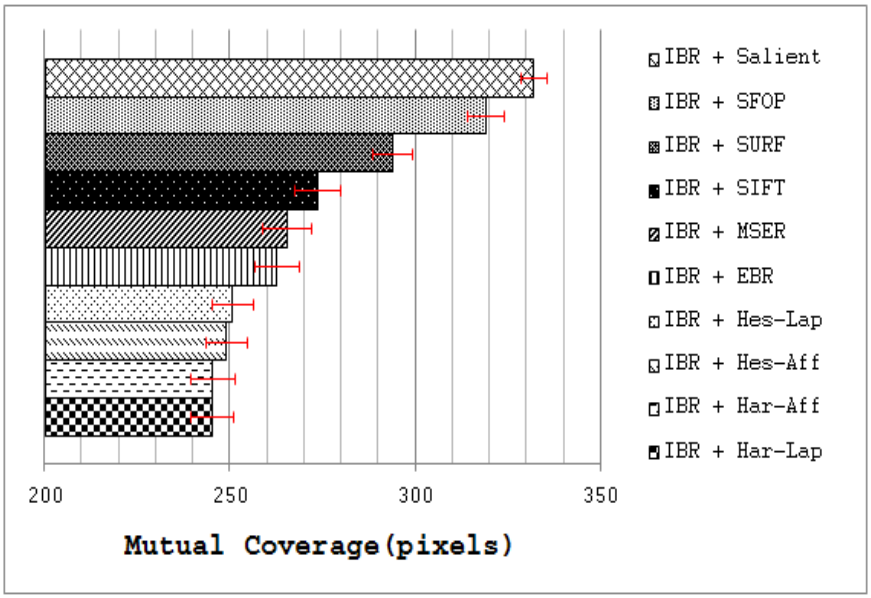

Figure 5-15: Mutual coverage of IBR in combination with other detectors for image database [192]; the error bars indicate the 95% confidence intervals for mean values

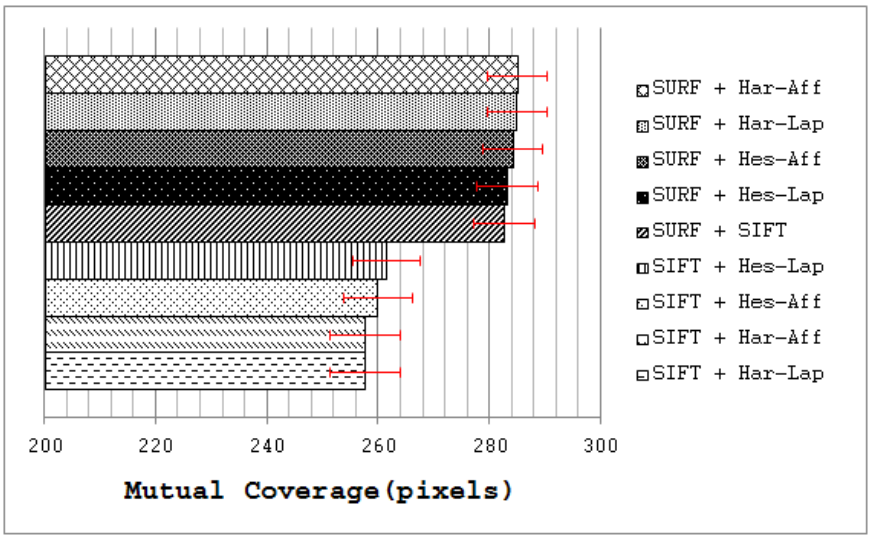

Figure 5-16: Mutual coverage of SIFT and SURF in combination with other detectors for image database [192]; the error bars indicate the 95% confidence intervals for mean values

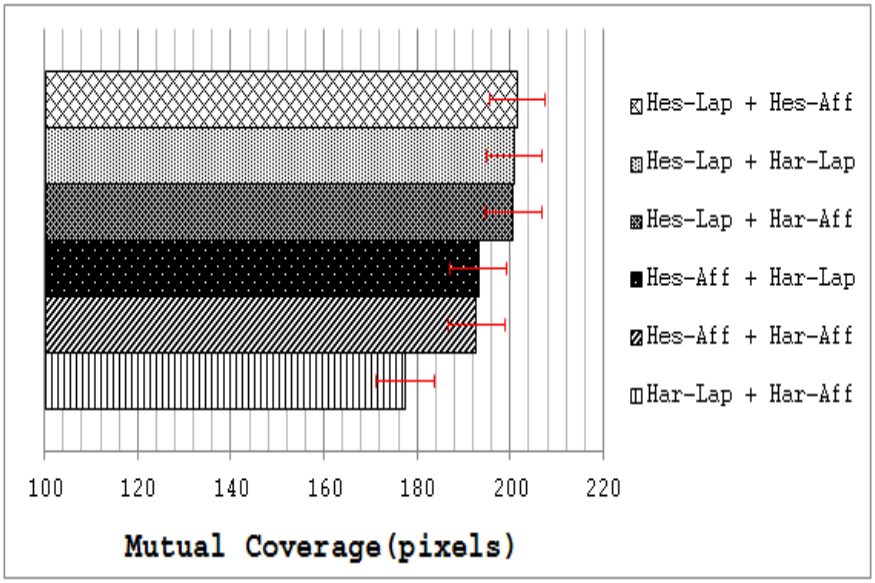

Figure 5-17: Mutual coverage of combinations of Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine for image database [192]; the error bars indicate the 95% confidence intervals for mean values

5.4.3 Results for Detector Triplets

In order to reduce the number of detectors to discuss for combinations of three, the results for detector pairs presented above are utilized for identification of possible similar trends in the behavior of detectors. This allows detectors showing similar characteristics to be grouped together. Some key inferences made from the results for detector pairs (Figure 5-11 to Figure 5-17) are described in the following paragraphs.

Although Salient is categorized as a blob detector in Table 5-4, its behavior is rather different from other detectors extracting the same feature type, such as SIFT and SURF. The author considers that this is in agreement with the underlying design principles of these detectors as Salient responds equally to different feature types whereas others show bias towards blobs. Salient is therefore separated from blob detectors and put into a new category of entropy-based detectors.

The behavior of MSER and IBR is similar when combined with all other detectors. Moreover, these two detectors achieve low coverage when grouped together. They are thus categorized as segmentation-based detectors (as in Table 5-4).

Although SURF and SIFT are both blob detectors, there are discrepancies in their behavior when combined with other detectors: For example, they provide similar performance when combined with a corner detector but different when grouped with a spiral detector. This disparity may be attributed to the method they use to detect blobs. SIFT approximates Laplacian using Difference-of-Gaussians whereas SURF is based on the determinant of the Hessian matrix. Although they do not complement each other well, as indicated by their relatively low mutual coverage (Figure 5-16), SIFT and SURF are placed in different categories as their behavior is inconsistent when combined with other detectors.

Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Affine exhibit similar behavior when combined with all other detectors. Low coverage values for combinations of these detectors indicate that they do not complement each other well. It is also evident that their behavior is different from Laplacian-based and Hessian matrix-based blob detectors. These detectors are therefore grouped together in a new category named 'hybrid' detectors which subsumes some detectors from the 'blob' category in Table 5-4 and others from the 'corner' category. Table 5-5 summarizes the re-categorization of the detectors under investigation.

Table 5-5: Re-classification of state-of-the-art detectors based on results for detector pairs

Category	Type	Detectors
1.	Laplacian-based	SIFT (Difference-of-Gaussians)
2.	Hessian Matrix-based	SURF (Determinant of Hessian)
3.	Hybrid detectors	Harris-Laplace, Hessian-Laplace, Harris-Affine, Hessian-Affine
4.	Corner detectors	Edge-based Regions (EBR)
5.	Spiral detectors	SFOP
6.	Entropy-based detectors	Salient
7.	Segmentation-based detectors	MSER, Intensity-based Regions (IBR)

By grouping detectors from three different categories in Table 5-5, the author has investigated the performance of detector triplets using image database [192]. Instead of presenting individual findings, the author has generalized the results for detector triplets and produced a ranking of these

combinations, which provides more useful insight into the performance of different detector categories in Table 5-5 when combined with other categories. Table 5-6 presents a rank-ordered list of those classes of detector triplets that achieve highest mutual coverage; it can be thought of as a guideline to choosing which classes of detector to combine. However, entropy-based detectors are slow to compute, making them undesirable for online use, the aim of this chapter, so Table 5-7 presents similar lists of detector triplet classes that exclude entropy-based ones. Indeed, two such lists are given: the first includes spiral (for detecting circular features) and the second excludes it (when circles are known not to be present in imagery).

It is evident from Table 5-6 that combining entropy-, spiral- and segmentation-based detectors produces the highest mutual coverage across all combinations of detector categories. For combinations that do not involve an entropy-based detector, grouping a spiral detector with a Hessian matrix-based and a segmentation-based detector provides the best performance. Combining a spiral detector with a segmentation-based and a corner detector also achieves good results. It is interesting to note that the Laplacian-based detector category does not appear in Table 5-7 due to the relatively low mutual coverages obtained; this is the same observation made in Table 3 of [19]. Overall, the results can be considered broadly consistent to the findings in [19]. In addition, these results provide a guideline as to which detectors to combine in applications that require a reasonable distribution of image features, such as image registration and accurate multi-view geometry estimation, apart from good repeatability and speed.

Table 5-6: Top ranking detector triplets in terms of detector categories

Rank	Detector Triplet
For all combinations	
1.	Entropy-based + Spiral + Segmentation-based
2.	Entropy-based + Spiral + Corner
3.	Entropy-based + Spiral + Hybrid
4.	Entropy-based + Corner + Segmentation-based

Table 5-7: Some other promising detector triplets in terms of detector categories

Rank	Detector Triplet
For combinations excluding Entropy-based detector	
1.	Spiral + Hessian Matrix-based + Segmentation-based
2.	Spiral + Corner + Segmentation-based
3.	Spiral + Hessian Matrix-based + Corner
4.	Spiral + Hessian Matrix-based + Hybrid
For combinations excluding Entropy-based and Spiral detectors	
1.	Hessian Matrix-based + Corner + Segmentation-based
2.	Hessian Matrix-based + Hybrid + Segmentation-based
3.	Hessian Matrix-based + Corner + Hybrid
4.	Hessian Matrix-based + Laplacian-based + Corner

5.5 Feasibility of Proposed Methods for Real-World Applications

This section discusses the viability of the proposed measures for real-world applications. It analyzes how well the results presented above map to real-world problems, both for detectors and their combinations. In particular, it shows that high coverage implies better performance for homography estimation. The section also provides a timing analysis that shows the speed of calculating coverage, allowing it to be employed online as part of a practical system.

5.5.1 Mapping Coverage Results to Practical Problems

Since the suitability of local feature detectors for automatic image orientation systems was studied in detail by [45] recently, it is interesting to compare the results of this work to those of [45]. That evaluation was done using SFOP, Entropy [45], SIFT, MSER, Harris-Affine and Hessian-Affine. For separate detectors, SFOP was identified as providing the overall best performance; SIFT and MSER work well with images having good and medium amounts of texture, whereas Harris-Affine and Hessian-Affine perform poorly. Although the author's results are obtained using a different database of images to [45], the conclusions drawn from the results of Section 5.3 largely agree with the findings in [45] as SFOP is recognized as the best among SIFT, MSER, Harris-Affine and Hessian-Affine. The coverage-based

performance measure ranks SIFT higher than MSER. Moreover, the quantitative evaluation of Section 5.3 also demonstrates that SIFT and MSER perform better on images with good and medium texture (Campus-1 and Campus-2 datasets in this case [192]) but their performance is somewhat poorer for images with low texture. Hessian-Affine and Harris-Affine are at the bottom according to the ranking, consistent with [45].

For detector pairs, it was concluded in [45] that combining Hessian-Affine with SIFT has a detrimental effect on performance for an automatic image orientation problem as they have highly redundant feature sets. The results for detector pairs in Section 5.4 also yield the same conclusion for a combination involving SIFT and Hessian-Affine. A combination of SFOP, SIFT and MSER was identified as the most promising setting in [45] for automatic image orientation; the author's results also identify this configuration as one of the top groupings when considering only those triple combinations that involve the detectors evaluated in [45]. The high degree of correlation between the results presented here and those of [45] provides evidence that coverage and mutual coverage provide reliable methods of determining spatial distribution of interest points for image feature detectors.

To illustrate the impact of these results on real-world applications, consider the task of homography estimation for the Leuven dataset [50]. The mean error was computed between the positions of points projected from one image to the other, using a 'ground-truth' homography from [50], and a homography determined using the above detectors. SFOP performed best, with a mean error of 0.245, whereas EBR achieved a poor value of 3.672, consistent with the results shown in Figure 5-2 and Figure 5-3. Figure 5-18 shows a plot of coverage (read values from the left ordinate axis) and mean homography estimation error (read values from the right ordinate axis) for the MSER detector utilizing the Bikes dataset [50]; this is a typical result. Pearson's correlation coefficient for the two curves is -0.90 with a p-value of 0.03, clearly indicating that a high coverage implies a low mean error of homography estimation.

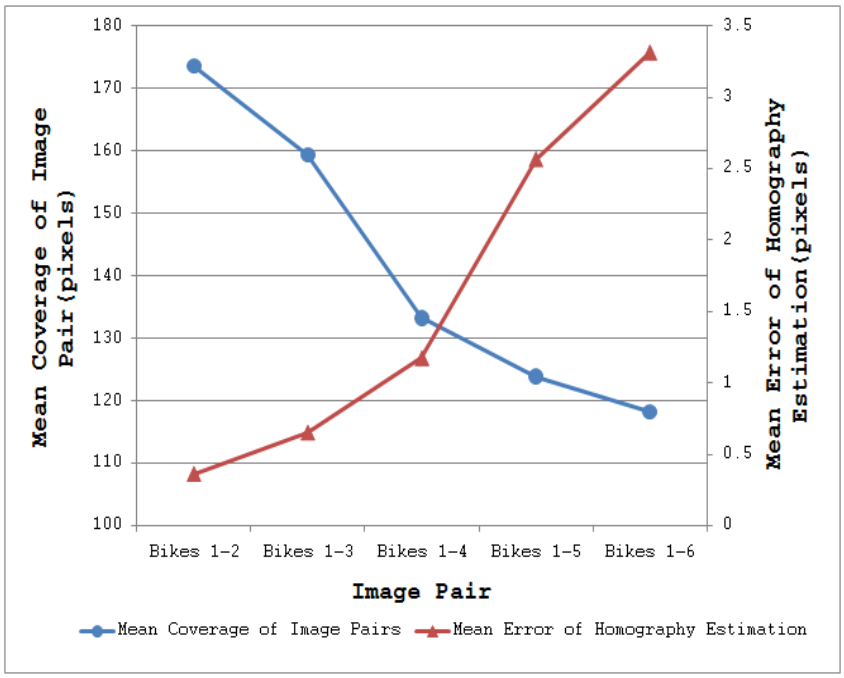

Figure 5-18: Curves for coverage and homography estimation error for MSER detector utilizing the Bikes dataset [50]

5.5.2 Computational Aspects

A method that can quickly predict the performance of feature detectors accurately would be valuable for time-critical applications. This section illustrates the potential of coverage and mutual coverage for ascertaining the performance of detectors and the complementarity of their combinations. Since the completeness measure [19] is to the author's knowledge the only existing scheme for carrying out such an analysis, coverage appears to be the first measure that makes possible the online adaption of feature detection to image content in order to improve performance.

Figure 5-19 and Figure 5-20 plot the total computation times for analyzing the performance of a specific detector and detector combinations respectively for 48 images of the Oxford datasets [50] utilizing coverage-based measures and the completeness measure of [19] (read values from the left ordinate axis). The dotted lines in these figures show the relative speed-up for the proposed methods as compared to the completeness tool (read

values from the right ordinate axis); the author has excluded the time taken to compute the reference entropy density for the completeness measure, some 716.68 minutes for the 48 images of the Oxford datasets. These results were obtained by running MATLAB implementations of these methods on a Linux-based HP ProLiant DL380 G7 system with Intel Xeon 5600 series processors. Since every detector extracts a different number of features for a given input image, as mentioned above, the mean number of interest points detected by every technique for the Oxford datasets is provided in Table 5-8 so as to visualize the dependence of computation time on the number of feature points.

It is evident from Figure 5-19 and Figure 5-20 that coverage has the potential to analyze feature detectors quickly. For example, analysis of the SFOP detector requires a mean time of only 241.85 ms per image. Detectors such as IBR, which have sparse feature sets, are analyzed more quickly (50.64 ms per image on average for IBR).

Table 5-8: Average number of interest points detected by state-of-the-art feature detectors for Oxford datasets [50]

	Bark	Bikes	Boat	Graffiti	Leuven	Trees	UBC	Wall
SIFT(DoG)	4549	1505	6939	4060	1910	10707	6310	11499
Salient	2238	2027	4231	2653	2081	5921	3817	6584
Harris-Lap	539	611	2107	2060	624	4669	1540	2520
Hessian-Lap	451	870	2527	3028	944	3942	1762	1479
Harris-Aff	537	590	2056	2041	612	4650	1500	2470
Hessian-Aff	450	801	2070	2424	757	3872	1617	1434
SURF(FH)	3526	2692	4822	5520	3405	7482	5184	5047
EBR	299	465	1024	1074	495	577	821	2716
IBR	706	673	635	807	330	1623	649	758
MSER	545	286	1012	692	392	2148	890	1975
SFOP	1735	1186	1692	1031	974	3159	1725	2720

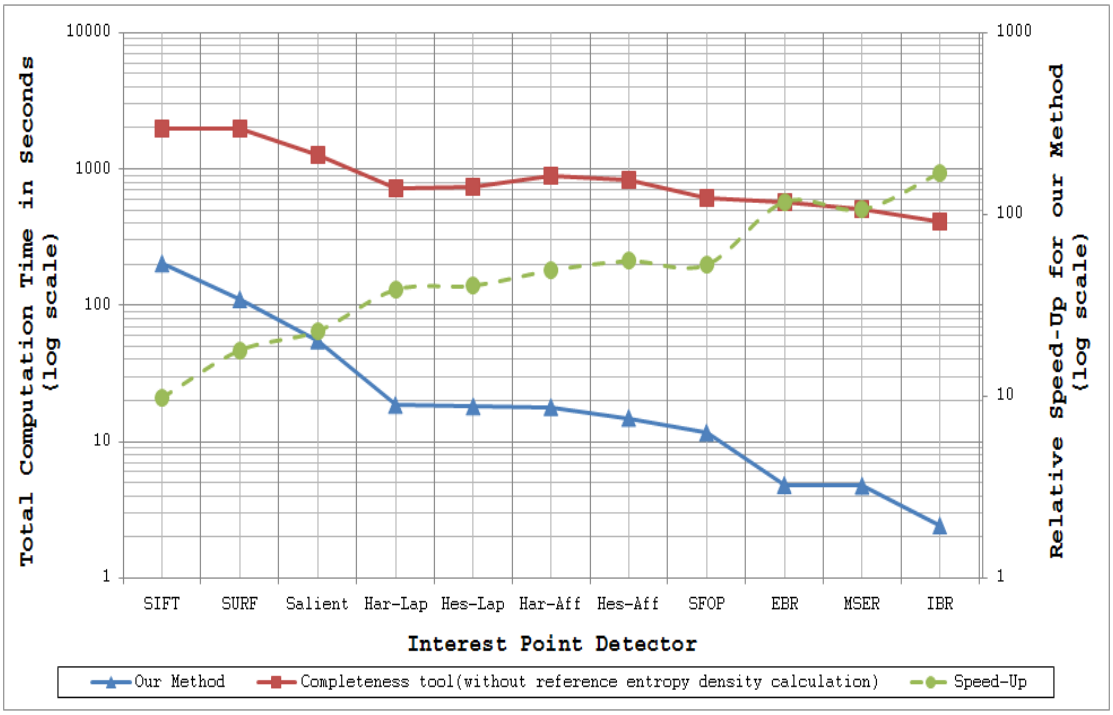

Figure 5-19: Timing analysis of the proposed coverage method and the Completeness tool [19] for 48 images of the Oxford Datasets [50]

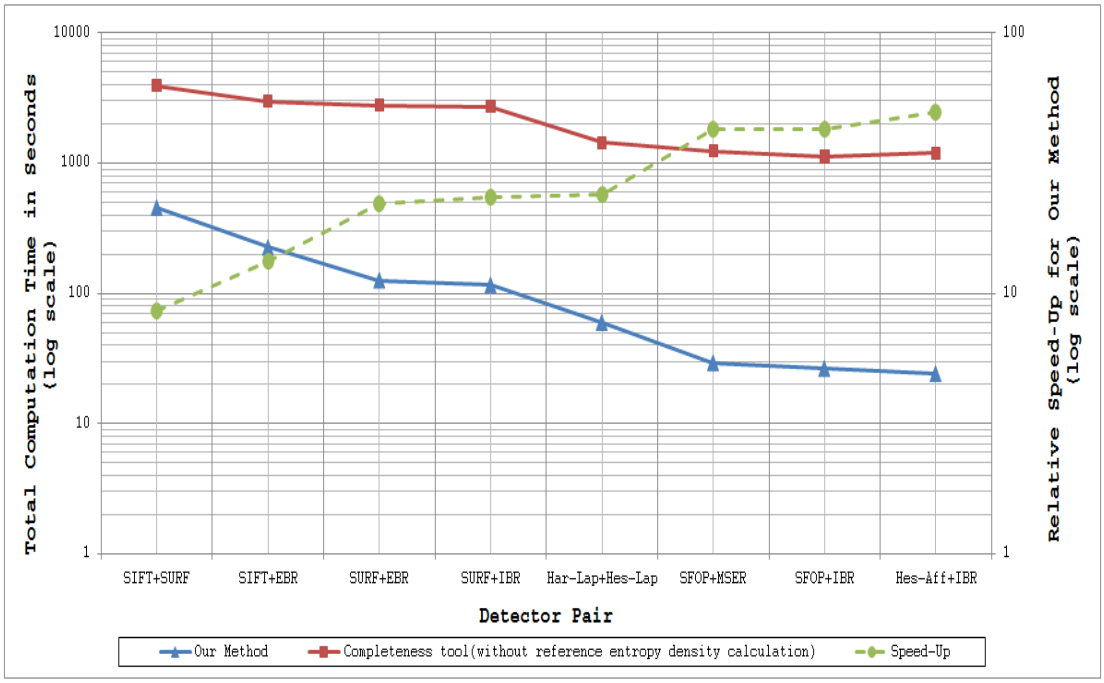

Figure 5-20: Timing analysis of the proposed mutual coverage method and the Completeness tool [19] for 48 images of the Oxford Datatsets [50]

5.6 A Prediction-based Framework for Combining Detectors

This section presents a principled framework for combining local feature detectors automatically, having the capability of handling varying scene types reliably, to achieve better performance in real-world applications that require a reasonable distribution of feature points. Utilizing the proposed framework, results are presented for the task of image registration which highlight its usefulness.

The emerging trend of running multiple feature detectors simultaneously to take advantage of complementary features for solving complex vision problems, such as category-level object recognition [44], stems from an inability to utilize different detectors in a selective and efficient manner depending upon the image content. Although this parallel approach may help in tackling the uncertainty of image content in situations where there is no prior knowledge available, it has detrimental effect on computation time due to increasing amount of data to process. Moreover, it results in an over-complete representation of an image rather than a compact one [1], and is not particularly useful for time-critical applications.

Complementarity of different feature types was first articulated in [187] which investigated the ability of edge- and blob-like features to carry image information based on a model of retinal cells for image reconstruction. With the aim of dealing with a wider range of images and exploiting several types of image structure, the desire to build an 'opportunistic' system by combining the output of several feature detectors was advocated by [122]. Similarly, a sparse texture representation using affine-invariant regions was proposed in [39] that utilized a combination of a corner and a blob detector. It details an interesting case study for which the recognition rate for a combination of detectors was lower than what was achieved using a single detector. This particular work emphasized two important points: the need to acquire a better understanding of the performance of different

detectors on different types of texture and to investigate how the output of different detectors can be combined so as to avoid detrimental effects on combined performance. Combinations of feature detectors have also been employed for category-level object recognition and object detection in videos [30, 43, 44]. As already mentioned in Section 5.5, the performance of different detector pairs and triplets was studied for the task of automatic image orientation in [45]. This work showed the negative effects on performance when SIFT is combined with Hessian-Affine and attributed it to the redundancy of features extracted by the two techniques.

The lack of a principled framework for combining feature detectors automatically in an effort to achieve better performance in real-world vision applications hence presents a major bottleneck. Development of such a framework is vital, as combining multiple detectors may have detrimental effects on combined performance, in some cases making it even lower than what can be achieved by a single detector [39, 45].

5.6.1 Proposed Framework

Figure 5-21 shows a block diagram of the proposed framework for combining local feature detectors automatically in vision applications that require a reasonable distribution of feature points. Depending upon the image content, the framework decides whether to operate in a single detector mode or employ multiple detectors. For predicting the performance of a single detector or a combination of detectors for a specific vision task, this framework utilizes the coverage and mutual coverage measures presented in Section 5.2.1 and Section 5.4.1 respectively. The aim here is not to produce an optimal solution (in the sense that it is the best conceivable) but rather to provide a reliable framework that allows performance to be improved when it is clear that a single detector will not perform adequately and to have a low enough overhead that it can be used online.

Before discussing the framework in detail, it is worth stating that the proposed framework is generic in the sense that it can be utilized for any set of local feature detectors and a variety of vision applications. To keep this

generality, the framework is discussed here without referring to any specific detector or giving example of any particular application; more specific results will be discussed later in Section 5.6.2.

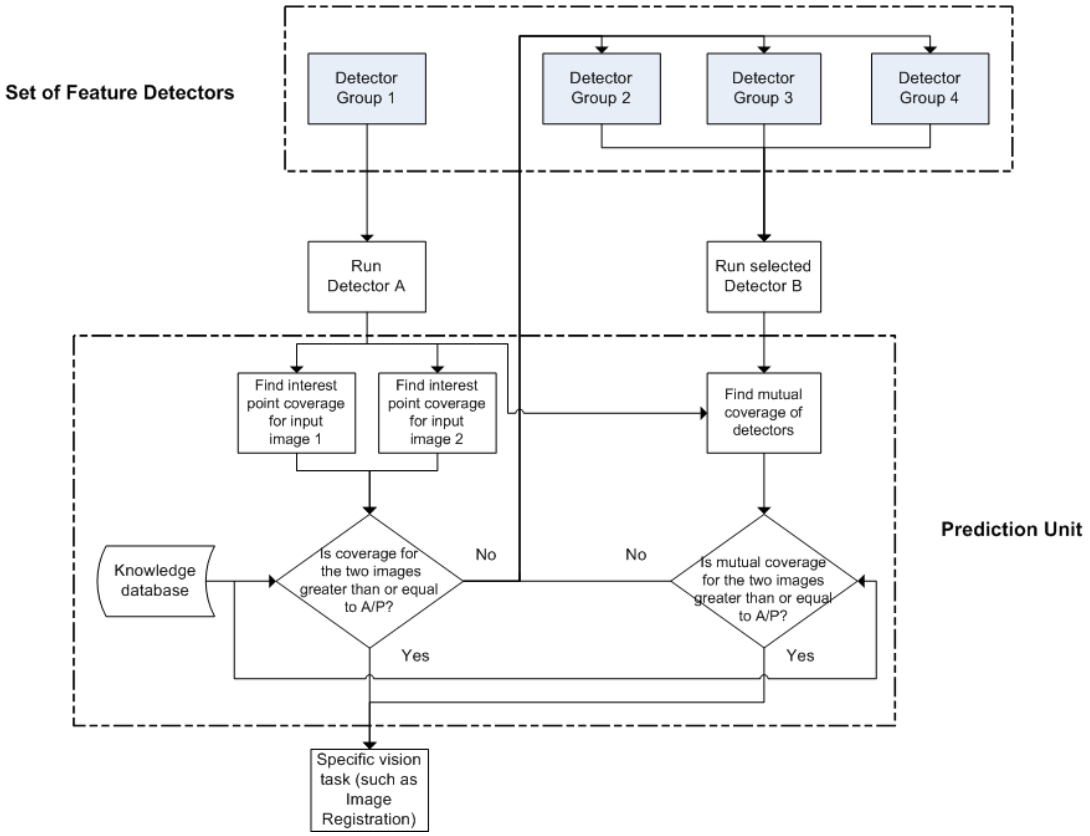

Figure 5-21: A block diagram of the proposed framework for combining local feature detectors

According to the proposed framework, the available feature detectors are first divided into specific groups based on general knowledge about complementarity of their detected features. For this categorization, the results given in Section 5.4, which provide a useful guideline for combining detectors in pairs and triplets, can be utilized. Any suitable detector is then selected from one of the groups to run on a pair of images. The coverage values are computed for the two sets of detected feature points utilizing the metric proposed in Section 5.2.1. A criterion is then needed to determine whether to use a single detector or a combination of detectors. As discussed in Section 5.3.3, the ratio of area to perimeter (Equation 5-5), which has long been used in physics for specifying field sizes [196], provides results

that are consistent with the visual inspections of Section 5.2 (see Figure 5-2 and Figure 5-3). It is therefore a suitable criterion to be used for ascertaining whether the coverage of a single detector is good enough. If the coverage values achieved by the selected detector are greater than or equal to the ratio of area to perimeter of image for both the images individually, the single detector mode is selected by the proposed framework and the rest of the processing required for the specific vision task (such as feature description and matching) is done utilizing the detected feature points.

In the event that the coverage value achieved by the selected detector for any one image is less than the ratio of area to perimeter of image, the proposed framework opts for multiple feature detectors for that particular image pair. For selecting another detector which can be combined with the first detector, a knowledge database is employed which contains information about the complementarity of different feature detector groups. Again, the results given in Section 5.4 can be utilized for building such a database. After getting the input from the knowledge database, a second detector is selected from a complementary detector group to the first; mutual coverage values are then calculated using the metric presented in Section 5.4.1 for both input images. If the computed mutual coverage values are greater than or equal to the ratio of area to perimeter of image, the detected feature points are selected and the rest of the processing is done. If this is not the case, the second detector is discarded and another detector is selected from some other detector group whose detected features are generally considered complementary for the first detector. This process of selecting a second detector is repeated until the required mutual coverage threshold is achieved for both the images. In case it does not happen after combining the first detector with all available detector groups, one of the earlier discarded detectors is used with the first detector on the basis that this combination yields the highest mutual coverage.

The proposed framework in Figure 5-21 can be extended in a number of ways. Instead of employing a pre-defined, fixed knowledge base, it is possible to utilize one which updates its stored information dynamically by

taking into account the current combined performance of different feature detectors. Another variation that can be introduced is to look for a third detector to make a triplet for the particular scenario when a detector pair fails to achieve the required mutual coverage values.

5.6.2 Results

To demonstrate the utility of the proposed framework, an image registration task is used here as it is dependent on achieving a reasonable spatial distribution of detected feature points. A database of 37 image pairs with rotation and viewpoint changes is employed for this particular task. Each image in the database has dimensions of 1080 x 717 pixels and any two images that form a pair have large overlapping regions, to provide ample opportunity for an employed detector to show its best performance. This database has been made available online at [197].

Figure 5-22: Image registration result for the image pair 7 of the database using IBR alone

Before presenting the results for the proposed framework, it is worth having a look at the individual performance of the detector to be employed as the starting detector for the framework. Here, IBR serves as the starting detector; although IBR manages to solve the image registration problem for

all image pairs in the database, there is large variation in the accuracy of registration. Figure 5-22 to Figure 5-25 show four sample registered image pairs from the database utilizing IBR alone.

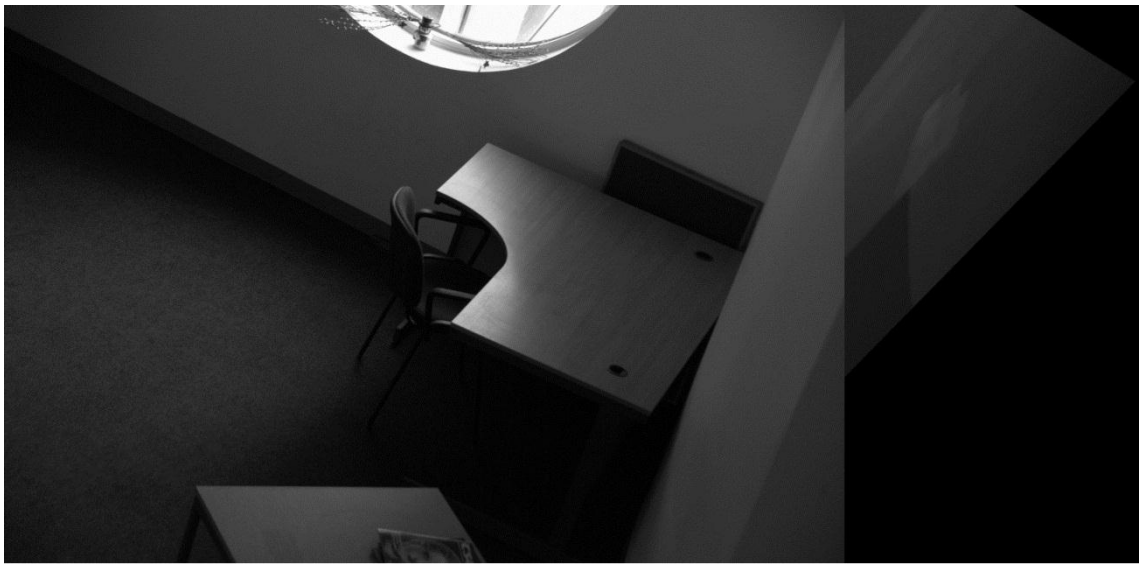

Figure 5-23: Image registration result for the image pair 8 of the database using IBR alone

It is evident from Figure 5-22 that image pair 7 is registered reasonably well from the feature matches of IBR. Contrary to that, the image registration result for image pair 8 is quite poor (see Figure 5-23). Although the results for image pair 4 (Figure 5-24) and image pair 12 (Figure 5-25) can be considered better than that of image pair 8, more accurate registration is desirable for these cases.

The variation in the accuracy of registration for the database when using feature points detected by IBR can be explained by the coverage values achieved by the detector for this database (as shown in Figure 5-26). It can be seen clearly that the coverage values of IBR for the image pair 7 are much greater than the area to perimeter ratio of image (215.45 for this particular case). The reasonable spatial distribution of detected features for both the images thus allows IBR to register this particular image pair accurately (Figure 5-22 and Figure 5-26). On the other hand, the coverage values for image pairs 4, 8, and 12 are below the required threshold of 215.45 and provide reasonable justification for the inaccurate registration

results shown in Figure 5-23 to Figure 5-25. It should be noted that coverage values for image 8 are particularly low, which ultimately leads to such a poor result.

Figure 5-24: Image registration result for the image pair 4 of the database using IBR alone

Figure 5-25: Image registration result for the image pair 12 of the database using IBR alone

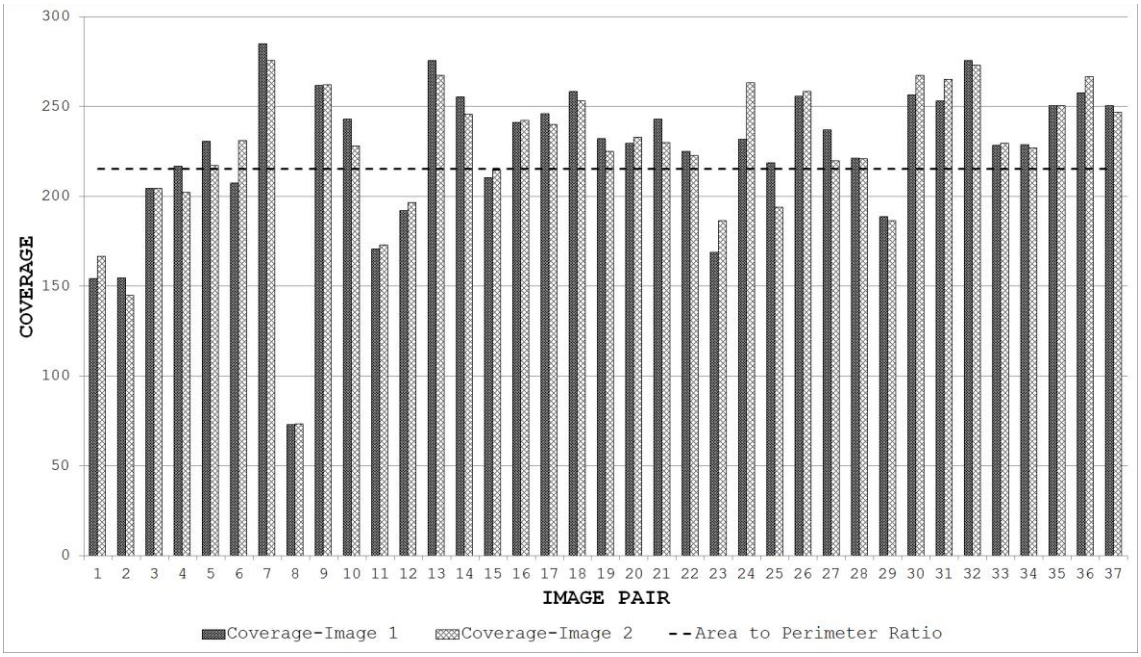

Figure 5-26: Coverage results of IBR for the database

When the proposed framework is employed with IBR as the starting detector (selected from the segmentation-based detector group), coverage values are computed for every image pair of the database as described in Section 5.6.1. The SFOP detector is then combined automatically with IBR for only those image pairs which have coverage values below the required threshold of area to perimeter ratio. For the remaining image pairs, the framework opts for the single detector mode (continuing with IBR only) as the coverage values are greater than or equal to 215.45. The coverage values achieved by this 'intelligent' dual mode system for the database are shown in Figure 5-27. To indicate when the framework selects single detector mode or employs multiple detectors, the operating mode is shown by numerical values in Figure 5-27. Note that the coverage values and the area to perimeter ratio should be read from the left ordinate axis whereas the value of operating mode should be read from the right ordinate axis. For the operating mode, a value of '0' indicates that the framework selects single detector mode for the current image pair, whereas a value of '1' shows that the framework employs multiple detectors for the specific image pair. It is clear that there is a marked improvement in the spatial distribution of

detected features as compared to the results shown in Figure 5-26. This improvement is apparent in the final output: Figure 5-28 to Figure 5-31 show the registration results for the four sample image pairs, and it is evident that all the image pairs are registered more accurately when the proposed framework is employed.

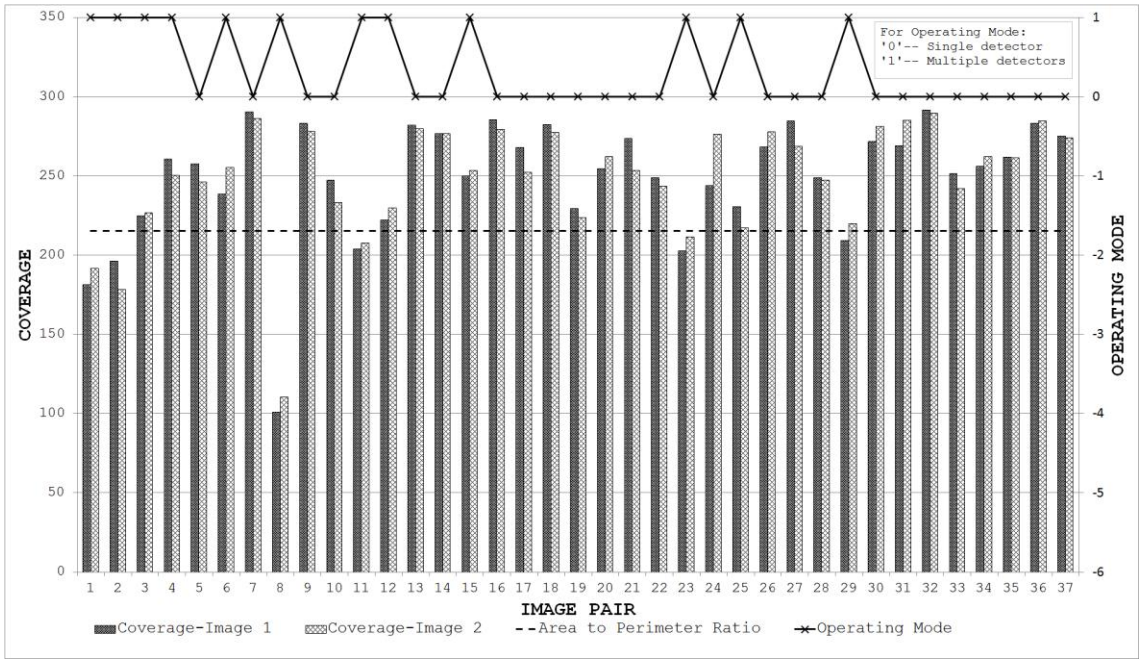

Figure 5-27: Coverage results achieved using the proposed framework for the database

Figure 5-28: Image registration result for the image pair 7 of the database using the proposed framework

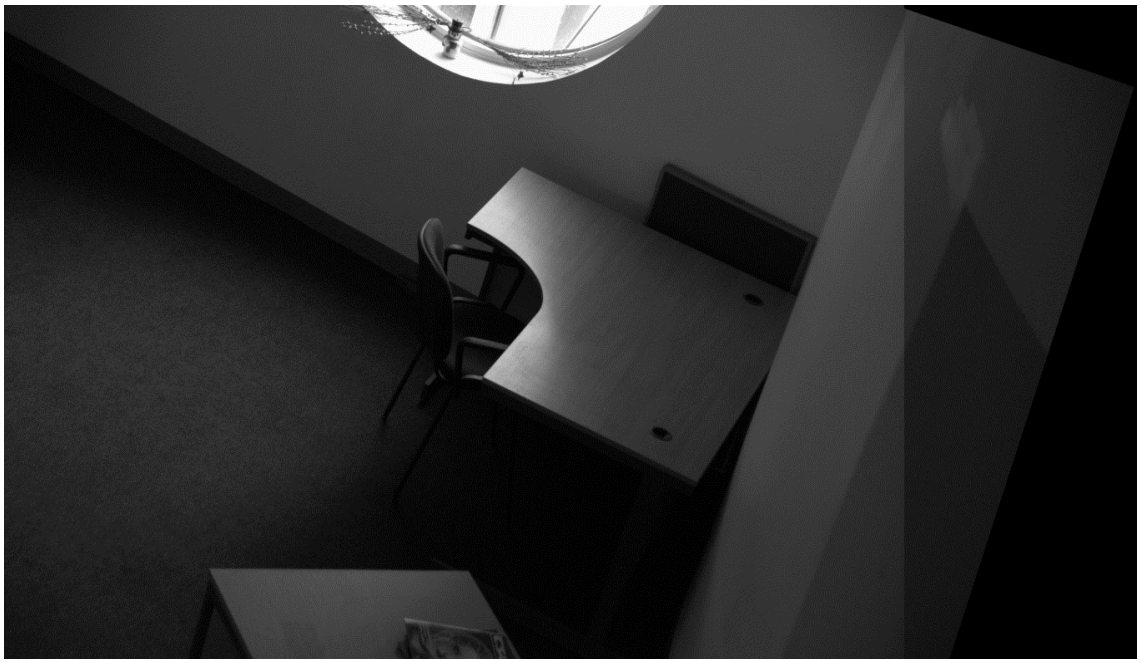

Figure 5-29: Image registration result for the image pair 8 of the database using the proposed framework

Figure 5-30: Image registration result for the image pair 4 of the database using the proposed framework

Figure 5-31: Image registration result for the image pair 12 of the database using the proposed framework

5.7 Summary

The spatial distribution of image features has received little attention until comparatively recently. This chapter has taken a step in this direction and presented a reliable method of measuring coverage which concurs with visual assessments. The proposed metric reflects the underlying principles of detectors and can be employed as a quick indicator of detector performance. It has also been found that the mutual coverage of several feature detectors, obtained simply by concatenating the feature points they detect and calculating the coverage of the combination, gives a rapid, principled way of determining whether combinations of interest point detectors are complementary without having to undertake extensive evaluation studies; indeed, calculation is so rapid that one can consider using it online in an intelligent detector that adds features from several detectors in order to ensure that coverage, and hence accuracy of subsequent processing, is good enough.

The chapter has presented coverage-based evaluation results for several state-of-the-art local feature detectors utilizing standard datasets. For quantitative analysis, a database of images containing both indoor and outdoor scenes with variations in texture was developed and a standard statistical test, McNemar's test, was employed to identify statistically-significant performance differences between detectors. The results obtained indicate the better performance of Salient and SFOP; other detectors, such as EBR, Harris-Laplace and Hessian-Laplace (and their affine-invariant versions) achieve a low ranking. The same image database was utilized to investigate detector pairs and triplets. Salient combined with SFOP provides the best performance in the case of detector pairs. Combining Salient or SFOP with a segmentation-based detector (IBR or MSER) also yields good coverage. For triplets, a segmentation-based detector or a corner detector added as a third component to the combination of Salient and SFOP is the most promising configuration. It is also identified that among combinations not involving Salient detector, grouping SFOP with a segmentation-based detector and SURF achieves high coverage. In an effort to provide a useful guideline for combining feature detectors in vision applications, the chapter has presented results for different detector classes.

It has been shown that, for detectors with known good repeatability, high values of coverage predict low errors in homography estimation, a task typical of a number of vision applications. Finally, the chapter has presented a prediction-based framework for combining local feature detectors in applications that require reasonable distribution of feature points.

6 An Algorithm for the Contextual Adaption of SURF Octave Selection with Good Matching Performance

Fast is fine, but accuracy is everything.

XENOPHON

Speeded-Up Robust Features (SURF) is a feature extraction algorithm designed for real-time execution, though this is rarely achievable on low-power hardware such as that in mobile robots. One way to reduce the computation is to discard some of the scale-space octaves, and previous research has simply discarded the higher octaves. This chapter shows that this approach is not always the most sensible and presents an algorithm for choosing which octaves to discard based on properties of the imagery. Results obtained with this *best octaves* algorithm show that it is able to achieve a significant reduction in computation without compromising matching performance.

6.1 Introduction

As already mentioned in Chapters 1 and 2, recent years have seen a great deal of effort expended within the research community towards techniques that are able to detect, describe and match image features [10, 12, 13, 17, 54, 55, 198-201]. The most popular of these algorithms operate in a way that makes them reasonably independent of scale and orientation changes between the images being matched. The technique known as SURF (Speeded-Up Robust Features) has a number of adaptions over earlier techniques such as SIFT (Scale Invariant Feature Transform) [11, 12] and the Harris-Laplace feature detector [200] that are intended to improve execution speed without compromising the effectiveness of feature detection [13, 53, 202]. Indeed, SURF has gained widespread popularity in vision systems due to its faster execution, paving the way for applications such as an interactive museum guide, retina mosaicking and mobile handheld augmented reality [203-205].

During the last decade or so, SIFT [12] has become the most popular technique for matching image features due to its fast detector coupled with a distinctive descriptor. However, the high dimensionality of this descriptor directly affects the amount of computation required for feature matching and is considered a major shortcoming of SIFT for real-time applications [13]. A number of optimizations and extensions have been proposed for the basic feature detection-description scheme presented by SIFT. The driving factors have been improvement in matching performance, reduction in amount of computation and adaptation for new applications. PCA-SIFT, GLOH, RIFT, CSIFT and n-SIFT represent some of the popular variants proposed to date [54, 55, 198, 199, 206]. PCA-SIFT applies principal component analysis (PCA) to the gradient around each detected interest point, coupled with reduction of the descriptor from 128 to 36 coefficients in an effort to speed up the subsequent matching phase. This, however, results in a less distinctive descriptor as compared to SIFT [55]. The GLOH descriptor uses more spatial regions in its histograms than SIFT but yields

the same number of coefficients by again employing PCA. Although the GLOH descriptor reputedly is more distinctive and robust than SIFT, this advantage is diminished by the extra computation required. Since color has the potential of providing valuable information in object description and matching tasks, *colored SIFT* (CSIFT) extends the basic algorithm by computing SIFT descriptors in a color invariant space; these have proven more robust to color and photometrical variations. Features from images of arbitrary dimensionality are extracted and matched by n-SIFT which is built upon the basic concepts outlined by SIFT. This method is particularly useful for automated matching of medical images such as MRI images.

SURF was devised using insights gained from previously-proposed feature detectors and descriptors. It employs a Hessian-based detector due to its documented higher stability and repeatability [15, 55]. Inspired by the DoG detector used by SIFT, it approximates the determinant of the Hessian matrix to detect blobs in an image. In addition, SURF utilizes an integral image representation [132] to reduce computation, leading to a significant speed-up [13]. As with most of the proposed descriptor schemes, SURF builds its 64-coefficient descriptor using the same basic strategy as SIFT but makes use of sums of Haar wavelet responses instead of gradient information. The resulting descriptor can out-perform the SIFT counterpart both in terms of computation speed and matching performance [13].

Despite being faster than contemporary techniques, SURF does not necessarily achieve real-time performance on modern desktop computers with software-only implementations due to its high computational complexity. For example, detection and description of 1,529 interest points using the original software implementation of SURF [207] for the first image (800 x 640 pixels) of the Graffiti data set provided by [50] takes about 610 ms on a standard Pentium-IV PC running at 3 GHz [13]. Since real-time performance is critical for vision-based applications such as target tracking and aerial surveillance, removing or reducing the bottlenecks that impede SURF from achieving real-time performance is desirable.

Recent research has targeted software-based optimization and/or hardware acceleration in an effort to improve the execution speed of SURF, with encouraging results. In [201], a speed-up by a factor of 2-6 relative to the original implementation [207] is achieved by running a multi-threaded software implementation of SURF on multi-core processors. Simulation results for a hardware-accelerated system are presented in [205] that demonstrate real-time performance, though for still images only. An implementation of SURF on programmable graphics hardware is detailed in [208] that can process image sequences at a rate of more than 100 frames per second. With the spectrum of embedded vision applications becoming broader and broader, there is enough motivation to investigate efficient software and/or hardware solutions, not only in terms of execution speed but also for computational resources, chip area, weight and power consumption.

Since computational complexity is the major bottleneck in achieving real-time performance, it is clear that algorithm-level optimization of SURF holds the key for faster software and/or hardware solutions. This chapter takes a step in this direction by exploring the reduction of scale-space 'octaves' as a key algorithm-level optimization for reducing the computational complexity of SURF. Conventionally, this is achieved simply by discarding the higher octaves; but the results presented below demonstrate that this may often reduce accuracy. Instead, this chapter develops and assesses a more sophisticated approach, termed *best octaves,* by selecting the optimal SURF octaves for a particular application; it will be demonstrated that this significantly out-performs the conventional approach both in terms of computation and matching performance. Unlike the original SURF algorithm, the *best octaves* method processes all four octaves at a uniform sampling rate of unity, thus providing a fair opportunity for all octaves to show their maximum performance. The proposed method then finds two octaves that provide the best matching performance according to criteria expounded below. To the author's knowledge, this is the first systematic approach to SURF octave selection.

The remainder of this chapter is structured as follows. Following a brief overview of the SURF algorithm in the next section, Section 6.3 examines the reduction of SURF octaves as a key method to improve execution speed. The above-mentioned image matching approach for SURF octave selection is presented in Section 6.4. A detailed statistical assessment of the proposed technique in terms of matching performance and reduction in computation is done in Section 6.5. Finally, a summary of the chapter is presented in Section 6.6.

6.2 An Overview of the SURF Algorithm

This section provides a brief overview of the SURF algorithm; see [13, 53] for an in-depth exposition. There are two main (and distinct) stages: detection and description. These are followed by feature matching, as shown in Figure 6-1. The key tasks performed at each stage are summarized below.

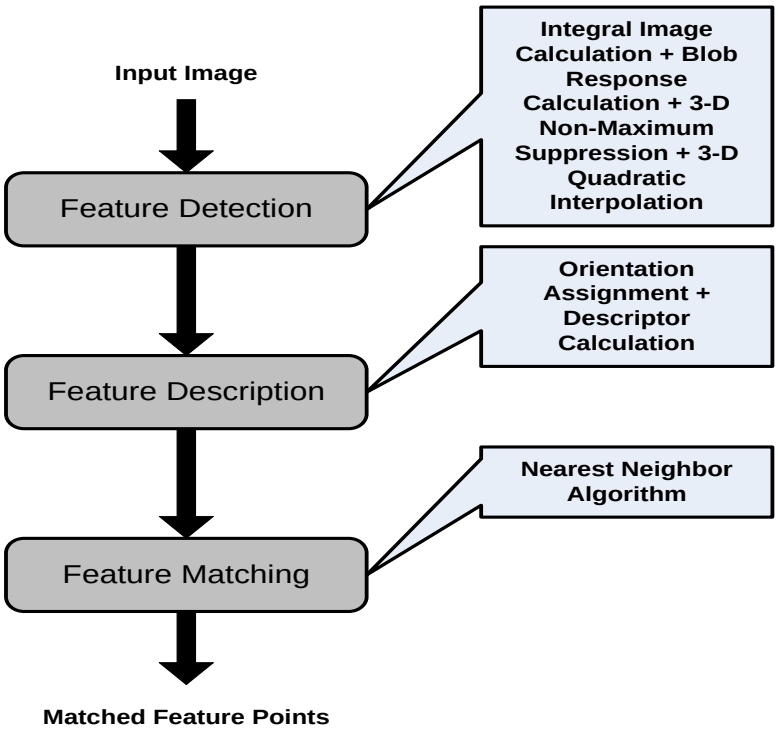

Figure 6-1: The key stages of SURF-based feature matching

6.2.1 Interest Point Detection

Interest point detection can be sub-divided into four steps: integral image calculation, computation of blob response maps at different scales, 3-D non-maximum suppression and 3-D quadratic interpolation. To eliminate the computationally-expensive multiplication operations when calculating box filters, SURF first computes the integral image representation for the whole input image, reducing the computation to three additions per pixel for calculating box filters, leading to significant speed improvements over other feature detection-description schemes. The value of the integral image at any location (*x,y*) in an image is the sum of all the pixels to the left and above it [132].

The next step is the calculation of blob response values for every image location at different scales. At any specific scale σ, the algorithm computes the blob response at an image location (*x,y*) by approximating the determinant of the Hessian matrix using:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \qquad \textit{Equation 6-1}$$

where D_{xx}, D_{yy} and D_{xy} are the convolutions of the input image with approximate second-order Gaussian partial derivatives in the *x*-, *y*- and *xy*-directions respectively, centered at location (*x,y*). As the second-order Gaussian partial derivatives are approximated as rectangular masks by SURF, increasing values of σ result in large mask sizes. By employing the integral image representation discussed above, SURF ensures efficient computation of D_{xx}, D_{yy} and D_{xy} at a constant speed, irrespective of the mask size. Once computed, these values of D_{xx}, D_{yy} and D_{xy} are normalized with respect to the mask size before being utilized in Equation 6-1.

To achieve scale-invariant detection of image features, a scale-space is constructed by convolving rectangular masks of increasing size, corresponding to different scales, with the input image. This procedure results in a series of blob response maps at different scales. The scale-space

is divided into a fixed number of octaves, where each octave consists of a specific number of adjacent scales at which the blob response maps are calculated. The maximum number of octaves that can be computed is therefore dependent upon the number of scales per octave and the highest scale at which blob response map can be calculated. The size of the input image determines this highest scale as the maximum size of mask must not exceed the input image size.

Normally, four scales per octave are used by SURF. With this configuration, four octaves are considered sufficient for scale-space analysis, as the contribution of octaves higher than that is negligible in terms of detected interest points [13]. Rectangular masks of size 9 x 9 are used for computation of the blob response map at the lowest scale ($\sigma = 1.2$). The blob response map for the next adjacent scale ($\sigma = 2.0$) is computed by increasing the size of rectangular masks by 6 pixels in total (15 x 15 masks). The same procedure is followed to compute blob response maps for other scales. In short, masks of size 9 x 9, 15 x 15, 21 x 21 and 27 x 27 are used for the first octave. For the second octave, SURF increases the size of masks by 12 pixels instead of 6 pixels (as in the case of first octave) and also doubles the sampling interval in the spatial domain. The motivation behind doubling the sampling interval is to reduce the computation time. Since interest points can be arbitrarily close together, this implies a loss of accuracy due to skipping every second pixel in the input image [12, 13]. The second octave thus utilizes masks of sizes 15 x 15, 27 x 27, 39 x 39 and 51 x 51 pixels. The size of masks for the third and the fourth octave is increased by 24 and 48 pixels respectively. In addition, the sampling interval for the third octave is doubled to 4, while for the fourth octave the sampling interval is 8, meaning that every fourth pixel is processed in the third octave and every eighth pixel in the fourth octave. The mask sizes for the third octave are 27 x 27, 51 x 51, 75 x 75 and 99 x 99 pixels. Similarly, the fourth octave is computed using masks of sizes 51 x 51, 99 x 99, 147 x 147 and 195 x 195 pixels.

Once the scale-space is constructed, local maximum detection (3-D non-maximum suppression) is performed. The highest and lowest scales in

every octave are used only for comparison purposes as the local maxima need to be detected in 3-D space. Thus, for the two central scales of every octave, the blob response value at any pixel is considered a local maximum in a neighborhood of 3 x 3 x 3 if it is greater than the blob responses of all 26 neighbor pixels. To speed up this computation, SURF employs a fast variant introduced by [133]. A blob response threshold is normally applied, so that only high-contrast interest points are selected.

The next step is 3-D quadratic interpolation of the detected local maxima, allowing SURF to achieve sub-pixel, sub-scale accuracy. As proposed by [134], a 3-D quadratic function is fitted to the interest points for finding the interpolated location of the local maximum.

6.2.2 Interest Point Description

Two key tasks are performed in this stage: orientation assignment and descriptor computation. Since scale-space analysis provides interest points that are only scale-invariant, a reproducible orientation needs to be identified for every detected interest point to make its descriptor invariant to image rotations, and both these stages are based around Haar wavelets. Firstly, Haar wavelet responses in the x and y directions are computed within a circular neighborhood of radius $6s$, where s is the scale at which the interest point was detected, around the interest point and the sampling step is set to s. To achieve a speed-up, SURF again utilizes the integral image representation of the input image, as computation of Haar wavelet responses requires convolution with rectangular wavelets having side length of $4s$. The calculated Haar wavelet responses are then weighted with a Gaussian ($\sigma = 2s$) centered at the interest point. Vectors of different lengths are obtained by summing all horizontal and vertical responses using a sliding orientation window of $\pi/3$. The orientation of the longest such vector is assigned to the interest point.

To compute the descriptor, the algorithm constructs a square region of size $20s$ aligned to the selected orientation and centered at the interest point. After dividing this square region into smaller 4 x 4 square sub-

regions, Haar wavelet responses are calculated for each sub-region in horizontal and vertical directions at 5 x 5 regularly-spaced sample points using wavelets with side length of 2*s*. Once computed, the wavelet responses are weighted with a Gaussian ($\sigma = 3.3s$) centered at the interest point. The horizontal and vertical responses are then summed for each sub-region. Finally, the 64-coefficient SURF descriptor is obtained by summing the absolute values of the horizontal and vertical responses for each sub-region, which is normalized to achieve contrast invariance.

6.2.3 Nearest Neighbor Matching

The final stage is image feature matching on the basis of computed descriptors. This can be achieved using a nearest neighbor strategy [12]. For matching interest points between a reference image and a test image, Euclidean distances are computed for a candidate interest point in the reference image with all interest points in the test image. If the ratio of the Euclidean distance of the first nearest neighbor to the Euclidean distance of the second nearest neighbor is greater than 0.7, a matched pair is detected [53]. To speed up the matching process, SURF utilizes the sign of the Laplacian (the trace of the Hessian matrix) as it helps to differentiate bright blobs on dark backgrounds from dark blobs on bright backgrounds. This implies matching interest points with the same contrast type.

6.3 Reducing the Number of SURF Octaves

As with most computer vision techniques, SURF is both computation- and data-intensive in nature. For example, integral image calculation is recursive in nature and involves a large number of additions, due to the processing of every pixel in the input image. On the other hand, orientation assignment requires evaluation of trigonometric functions, which is also computationally expensive. Thus, to be able to improve the speed of SURF, one needs an understanding of the effect of the various parameters on each of its stages. Table 6-1 summarizes how the various stages scale, where n is

the image resolution, m is the number of detected local maxima, i is the number of detected interest points and k is the product of the number of feature descriptors for a test image and the number of feature descriptors for a reference image. The computation time of the detection stage is dependent upon image resolution and has a near linear relationship; for example, on an Intel Atom Platform running the OpenCV implementation of SURF [209] under Windows at 1.6 GHz, increasing the image resolution by a factor of 1.56 from 640 x 480 to 800 x 600 pixels results in an increase in computation time of the detection stage by a factor of 1.57, from 183 ms to 287 ms; thus, increasing the number of pixels in the image increases the computation time for the detection stage linearly [210]. The computation time of the feature description phase is a function of the number of detected interest points: the higher the number of detected interest points, the longer the computation time. On average, 0.2 ms per interest point is required to compute a SURF descriptor on a Pentium IV clocked at 3 GHz using the original implementation of SURF [202]. Finally, the computation time of the matching stage is determined by the number of feature descriptors for the test image and the number of feature descriptors for the reference image or in a feature database.

Table 6-1: Computational complexity of SURF-based image matching

S.No.	Stage	Computational Complexity
Detection		
S1.	Integral Image Calculation	$O(n)$
S2.	Blob Response Calculation	$O(n)$
S3.	3-D Non-Maximum Suppression	$O(n)$
S4.	3-D Quadratic Interpolation	$O(m)$
Description		
S5.	Orientation Assignment	$O(i)$
S6.	Descriptor Calculation	$O(i)$
Matching		
S7.	Nearest Neighbor Algorithm	$O(k)$

Reduction of the number of octaves below the usual four should reduce the computation time (and also reduce utilization of other resources, such as memory) for each stage of the algorithm and hence the focus in this chapter is on optimization of the number of octaves required for any particular vision application, though this must not be achieved at the expense of accuracy. This approach will benefit both software and hardware solutions.

6.3.1 The Conventional Approach

A technique that is commonly used for reducing the number of SURF octaves is simply to discard higher octaves in favor of lower ones: the lowest octave (octave 1) has the highest preference and the highest octave (octave 4) has the lowest preference. For example, only the first three octaves are processed in [201] for an image size of 1024 x 768 pixels.

For the purpose of discussion and to demonstrate the effectiveness of this approach in reducing the computational complexity, the first and the second image of the Boat data set [50] are utilized here – see Figure 6-2 and Table 6-2. With four octaves and a blob response threshold of 0.002, some 2,360 interest points are detected for the first image and 2,500 interest points for the second image using the OpenSURF implementation [211]. When the first image is matched with the second image, some 210 point correspondences are obtained using the nearest neighbor algorithm. For the case when only the lower two octaves (*i.e.,* 1 and 2) are processed with the same threshold, the number of detected interest points for the images are 2,050 and 2,227 respectively and the total number of point correspondences between them is 177. Hence, there is a decrease of 15.7% in the number of matched points when the number of octaves is reduced from four to two; this can be considered negligible. Processing only the lower two octaves and rejecting all the higher octaves is therefore a fair compromise between computational complexity and matching performance in this particular case.

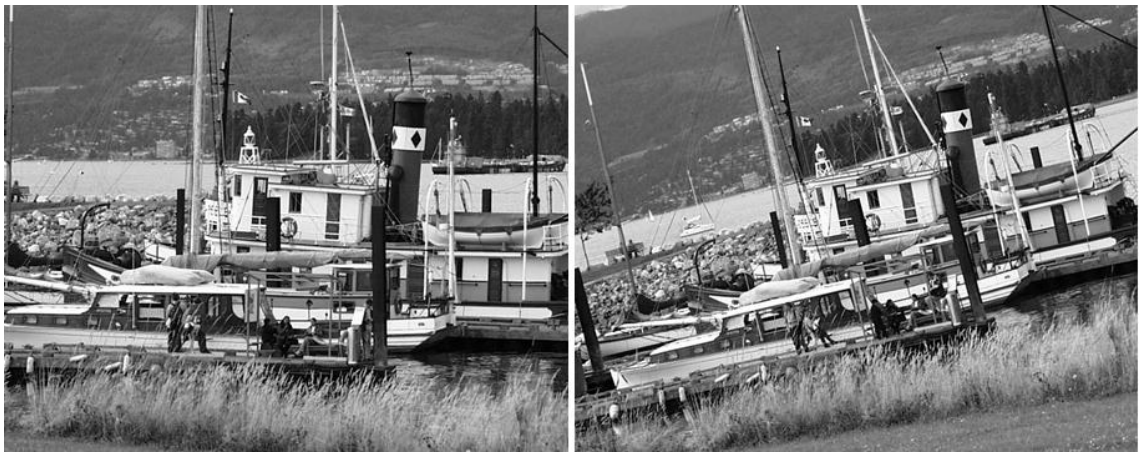

Figure 6-2: The first (left) and the second image (right) of the Boat data set [50]

Table 6-2: Results for the first and the second image of the Boat data set

Image	Octaves	Scales Per Octave	Threshold	Interest Points	Matches	Performance Decrease
1	1 2 3 4	4	0.002	2360		
2	1 2 3 4	4	0.002	2500	210	--
1	1 2	4	0.002	2050		
2	1 2	4	0.002	2227	177	15.7%

6.3.2 Limitations

The obvious limitation of the method discussed above is that it can be used to reduce the number of octaves only for images where the contribution of higher octaves is negligible. The selection of lower octaves and rejection of higher octaves for processing is actually based upon the assumption that the lower octaves always detect more interest points than higher octaves [13]. However, the author has observed that this assumption is not always true in real-life vision applications as there is a strong possibility of higher octaves being more dominant than the lower octaves in terms of detected interest points once the blob response threshold is applied. Thus, applying this approach to images where higher octaves are more significant in terms of detected interest points may introduce a dramatic degradation in image matching performance, which is certainly not desirable. Simply discarding

the higher octaves would also result in failure if higher octaves are less dominant in terms of detected interest points but more significant in terms of matched interest points compared to lower octaves.

Figure 6-3: The fifth (left) and the sixth image (right) of the Bikes data set [50]

To illustrate the adverse effects of this kind of octave reduction, two sample cases are discussed here. The first uses the fifth and the sixth images of the Bikes data set [50] (Figure 6-3). The image matching results for the two images with four and two octaves are detailed in Table 6-3, again using OpenSURF [211]. It is evident from Table 6-3 that there is a considerable decrease (58.3%) in the number of matched interest points when the number of octaves is reduced from four to two. This degradation of image matching performance is certainly not desirable and is a sharp contrast to the results presented in Table 6-2. Simply discarding higher octaves is therefore not a sensible approach for this particular case.

Table 6-3: Results for the fifth and the sixth image of Bikes data set

Image	Octaves	Scales Per Octave	Threshold	Interest Points	Matches	Performance Decrease
5	1 2 3 4	4	0.0022	101		
6	1 2 3 4	4	0.0022	74	36	--
5	1 2	4	0.0022	59		
6	1 2	4	0.0022	30	15	58.3%

Figure 6-4 shows the second sample case which consists of two images from an aerial sequence. The image matching results for these two images using OpenSURF are presented in Table 6-4. With four octaves, there are 59 interest point matches, whereas with two octaves there are only 21. The drastic reduction of 64.4% in image matching performance here also demonstrates the detrimental effect that simply discarding higher octaves can have.

Figure 6-4: The 47th (left) and the 48th image (right) of an aerial sequence

Table 6-4: Results for the 47th and the 48th image of an aerial sequence

Image	Octaves	Scales Per Octave	Threshold	Interest Points	Matches	Performance Decrease
47	1 2 3 4	4	0.0022	349		--
48	1 2 3 4	4	0.0022	278	59	
47	1 2	4	0.0022	202		64.4%
48	1 2	4	0.0022	144	21	

6.4 Proposed Method

To overcome the limitations of the conventional method, this section presents a technique for SURF octave selection, which the author has termed the *best octaves* method. This section presents a comparative analysis of the best octaves approach with the conventional one, both in terms of matching performance and computation.

6.4.1 Underlining Principles

Since the performance of any SURF-based vision system relies heavily on the number of matched points, it is essential to keep the number of matched points as high as possible. Moreover, assessment based around matched points is preferable to one that considers only detected interest points as it is matches that determine the accuracy of the vision system: detection of a large number of interest points does not guarantee a large number of interest point matches. Conversely, it is entirely possible to obtain a significant number of interest point matches from a small group of detected interest points. Rather than preferring lower octaves based on the assumption that they detect more interest points, the focus here is on obtaining the maximum number of interest point matches without preferring any particular octave, while keeping the computational cost as low as possible.

Before detailing the main steps of the proposed method, let us consider briefly the significance of the sampling rate during the detection phase. The developers of SURF recommended doubling the sampling interval in the spatial domain when moving from lower octaves to higher octaves during calculation of the blob response map and other detection stages in order to reduce computation [13]. More precisely, all scales in the first octave are processed at sampling rate of 1 whereas, for the second, third and fourth octaves, the sampling rate is 2, 4 and 8 respectively. To reduce computation further, sampling rates of 2, 4, 8 and 16 respectively can be used for the four octaves, though the numbers of detected and matched interest points are reduced. This non-uniform sampling of octaves implies that the first octave has the highest preference whereas the fourth octave has the lowest preference; it also indicates that it is presumed that the lowest octave will detect the highest number of interest points and the highest octave the fewest. The above results demonstrate that this assumption is not always true and, once the blob response threshold is applied during the detection phase, any of the four octaves can contribute

more detected interest points and matched interest points. In summary, the non-uniform sampling strategy employed by SURF does not provide an equal opportunity to all the octaves to show their maximum performance in terms of detected and matched interest points. It is interesting to note that if the number of scales per octave is increased, some of the scales move from the higher octaves to the lower octaves and are processed at double their previous sampling rate. This again shows the inclination of SURF towards lower octaves, because if a particular scale is considered to detect a lower number of interest points due to the application of a large Gaussian filter and is thus sampled at a lower rate to reduce computation, it should be sampled at the same rate irrespective of the fact whether it is in the highest or the lowest octave.

Since detected interest points can be arbitrarily close together in the image, this non-uniform sampling inevitably incurs some loss of accuracy [12, 13]. This can be overcome by sampling all four octaves uniformly at a sampling rate of unity. It thus provides the maximum performance configuration in terms of detected and matched interest points at a particular blob response threshold. A comparison of the computation involved in this four-octave maximum performance configuration with non-uniformly sampled variants for the first three stages of SURF is shown in Figure 6-5. The computation for the first stage (S1) is equal for the four configurations, thus leading to zero percentage reduction as depicted in Figure 6-5. However, the maximum performance configuration involves significantly more computation compared to the non-uniformly sampled alternatives in stages S2 and S3. It should be noted that the reduction in computation with respect to the maximum performance configuration shown in Figure 6-5 is independent of the images being matched.

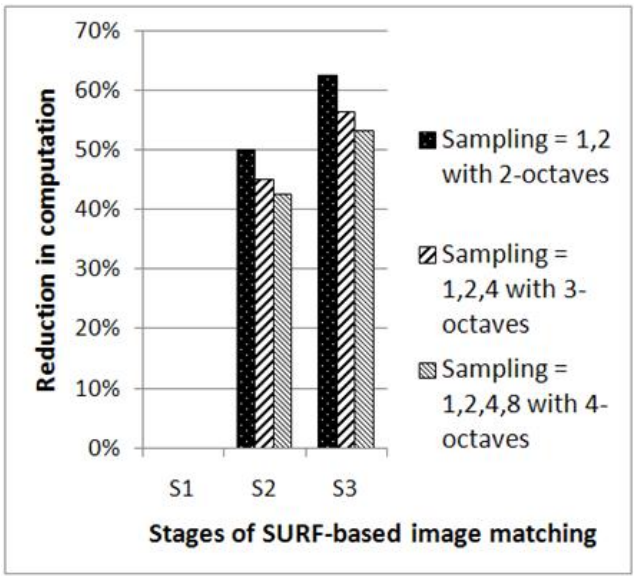

Figure 6-5: Comparison of computation for stages S1, S2 and S3 between maximum performance and non-uniformly sampled SURF configurations

The amount of computation for the next four stages depends upon the particular images being matched. The results presented in Table 6-2 to Table 6-4 are obtained using sampling intervals of 1, 2, 4 and 8 for the four octaves respectively. For the purpose of discussion, the 47th and the 48th image of the aerial sequence in Figure 6-4 are utilized here and the results are presented in Table 6-5. The results for the maximum performance configuration in Table 6-5 are obtained by a modified version of OpenSURF for achieving uniform sampling rate of unity. The maximum performance configuration provides 135 interest point matches (Table 6-5) as opposed to the 59 interest point matches (Table 6-4) provided by the non-uniformly sampled configuration at a threshold of 0.0022. Thus, the matching performance in this particular case of the non-uniformly sampled approach with four octaves is only 43.7% of the performance of the four-octave configuration processed at uniform sampling rate of unity. This is a significant reduction in matching performance, and trying to reduce it further by the conventional method of discarding octaves compromises accuracy.

Since the number of matches for the non-uniformly sampled SURF with four octaves cannot be increased beyond 59 at threshold of 0.0022 in

this particular case, the threshold can be lowered to 0.0009 (Table 6-5) to find an equal number of matched points. Although lowering the threshold too much can reduce the quality of interest point matches due to noise and hence is not desirable in practice, it has been done here to make a fair comparison with the maximum performance configuration in terms of computation. It is evident from the results presented in Table 6-5 that, for achieving the same number of matches, the non-uniformly sampled system with four octaves requires processing 44.4% more interest points compared to the system with a sampling rate of unity. Alternatively, we can say that 10.4 interest points are processed per match for the system with unity sampling rate, whereas 19.0 interest points need to be processed per match for non-uniformly sampled system with four octaves. A comparison of the computation involved in the last four stages of SURF-based image matching is shown in Figure 6-6; it is apparent that the maximum performance configuration provides more performance with less computation in this particular case, compensating for the extra computation done in the first three stages of algorithm (Figure 6-5), and thus out-performs the non-uniformly sampled configuration in terms of both performance and computation. The unity sampling rate configuration provides the maximum matching performance but its potential to out-perform the non-uniformly sampled configuration in terms of computation is more dependent upon the specific images being matched.

Table 6-5: Results for the 47th and the 48th image of an aerial sequence with sampling interval = 1 and sampling interval = 1, 2, 4 and 8

Image	Octaves	Scales Per Octave	Threshold	Sampling Interval	Interest Points	Matches	Interest Points Processed Per Match
47	1 2 3 4	4	0.0022	1	767		
48	1 2 3 4	4	0.0022	1	648	135	10.4
47	1 2 3 4	4	0.0009	1 2 4 8	1317		
48	1 2 3 4	4	0.0009	1 2 4 8	1231	134	19.0

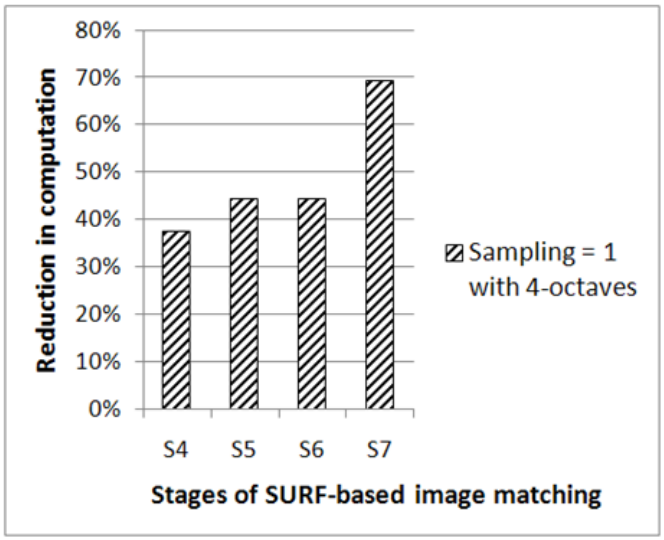

Figure 6-6: Comparison of computation for stages S4, S5, S6 and S7 between maximum performance and a non-uniformly sampled, 4-octave SURF configuration having equal performance at lower threshold for aerial images

To summarize, this discussion has underlined two basic principles for the selection of SURF octaves: equal opportunity for all the octaves to show their maximum performance; and to prefer the octaves with the best performance irrespective of being a lower or a higher octave.

6.4.2 The Best Octaves Approach

In *best octaves*, all four octaves are uniformly sampled to provide a fair opportunity for all octaves to show their maximum performance and to allow better evaluation of their relative performance. The proposed method then finds the two octaves that provide the best matching performance. For any given pair of images, the main steps of the proposed method are outlined in the following paragraphs.

Step 1. The matched interest points are calculated for the two given images for the maximum performance configuration (unity sampling rate and four octaves) and the number of them is considered as a reference for the later steps.

Step 2. The matched interest points are calculated for the two given images with unity sampling rate and the first two octaves. The ratio of the number of matched points, R, for octaves 1 and 2 against the reference is computed

to assess the effect of octave reduction. If $R \geq 0.5$, more than half the matches lie in octaves 1 and 2, so they are selected. Otherwise, we proceed directly to step 4, omitting step 3. A threshold of 0.5 is chosen here because a much lower value of R (say 0.25) may make the performance of the selected octaves only 25% of that of the maximum performance configuration in some cases, which is undesirable. Conversely, if R is much larger than 0.5 (say 0.85), the probability of obtaining that proportion of matches within two octaves becomes small. The experimental results presented later in this chapter show that this approach does not reduce the number of true matches; rather, it appears to reduce the number of false matches, thereby improving overall performance.

Step 3. Since the first two octaves are sampled at rates of 1 and 2 respectively in the original SURF algorithm, sampling every pixel of octave 2 for the selected best octaves in step 2 above may not make a significant difference to matching performance. To ensure that no extra computation is done, the matched interest points for the two images are calculated for the first two octaves with sampling rates of 1 and 2. The number of matched points for this non-uniform sampling case is compared with the reference; if greater than 0.4, then a non-uniform sampling rate of 1 and 2 is chosen for the selected best octaves in step 2 above. Otherwise, unity sampling rate is selected and the next three steps are skipped.

Step 4. The matched interest points are calculated for the two images with unity sampling rate and octaves 2 and 3 only. The Gaussian filters applied at different scales of octaves 2 and 3 are the same as in the original SURF algorithm. Similarly, after discarding octaves 1 and 2, the number of matched points is computed for octaves 3 and 4 by applying Gaussian filters to different scales, as in the original algorithm. The ratio of the number of matched points to the reference is then computed for the two cases and compared with each other to determine which is greater. If the maximum ratio is ≥ 0.5, then octaves corresponding to that ratio are selected as the best octaves. A sampling rate of unity is chosen and the next two steps are skipped.

Step 5. For the two images, the matched interest points are computed with unity sampling rate for each of octaves 1 and 3, octaves 1 and 4, and octaves 2 and 4. The ratio of the numbers of matched points to the reference is then computed for the three cases and compared to determine which is greatest. If the maximum ratio is ≥ 0.5, then octaves corresponding to that ratio are selected as the best octaves. A sampling rate of unity is chosen and the next step is skipped.

Step 6. Finally, the maximum of the six ratios calculated for octaves 1 and 2, octaves 2 and 3, octaves 3 and 4, octaves 1 and 3, octaves 1 and 4, octaves 2 and 4 with unity sampling rate is determined and the octaves corresponding to the maximum ratio are selected as the best octaves. A sampling rate of unity is chosen for the best octaves except for the case when octaves 1 and 2 are selected, in which case the sampling rate is determined as described in step 3.

It should be noted that, due to low frame-to-frame motion in the case of image sequences with medium to high frame rate, the number of matches in Step 1 need to be computed for the first pair of images only, then utilized as reference for the next few images based on frame rate, and then updated.

6.4.3 Qualitative Results and Comparative Analysis

To demonstrate the effectiveness of the proposed *best octaves* method, results for three sample cases are presented here and a comparative analysis is performed with the reference configuration (unity sampling interval and four octaves) and non-uniformly sampled SURF systems in terms of both matching performance and computation. The non-uniformly sampled variants that are being utilized here for comparison with the best octaves method are: SURF with 4-octaves (this is effectively the full version of the algorithm [13]); SURF with 3-octaves; and SURF with 2-octaves. For evaluation of matching performance, it is important to consider not only the total number of nearest neighbor matches but also their quality in terms of true matches. Different methods, such as Precision-Recall curves and ROC curves, can be employed to determine the quality of matches after their

categorization into true positives, false positives, true negatives and false negatives [55]; here ROC and Sensitivity-Specificity curves are used to demonstrate the quality of matches for the algorithms being evaluated, and support this with a quantitative statistical analysis in Section 6.5. The results have been obtained using OpenSURF modified for unity sampling rate at a threshold of 0.0022.

6.4.3.1 Matching Performance

Aerial images 47 and 48 (Figure 6-4) are again utilized here as the first sample case. Figure 6-7 shows the results of the *best octaves* method for these aerial images; the bars in the figure represent the number of points (read values from the left ordinate axis) whereas the line graph shows the matching ratio (read values from the right ordinate axis). Octaves 3 and 4 are selected as the best octaves in this particular case as the first two octaves have less than 50% matches as compared to the reference. It should be noted that with the selected best octaves, the decrease in matching performance with respect to the reference is less than 30%, a sharp contrast to the results achieved with non-uniformly sampled SURF (Table 6-4). To illustrate the performances of the *best octaves* method and other SURF configurations in this particular case, a Receiver Operating Characteristic (ROC) curve is shown in Figure 6-8, where the true positive and false positive rates were calculated using [212]:

$$True\ Positive\ Rate = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad \textit{Equation 6-2}$$

$$False\ Positive\ Rate = \frac{False\ Positive}{False\ Positive + True\ Negative} \qquad \textit{Equation 6-3}$$

It is clear that the best octaves method out-performs the others.

Results of the best octaves method for the first and the fifth image (Figure 6-9) of the Trees data set [50] are presented as the second sample case in Figure 6-10. As in Figure 6-7, the bars in Figure 6-10 represent the

number of points (read values from the left ordinate axis) whereas the line graph shows the matching ratio (read values from the right ordinate axis). Here, octaves 2 and 3 are selected as the best octaves and provide a matching performance only 31.4% less than the reference configuration. Again, to assess the quality of matches, a comparison of ROC curves for these images is presented in Figure 6-11. Clearly, the best octaves approach out-performs all the other SURF configurations comprehensively, including the reference.

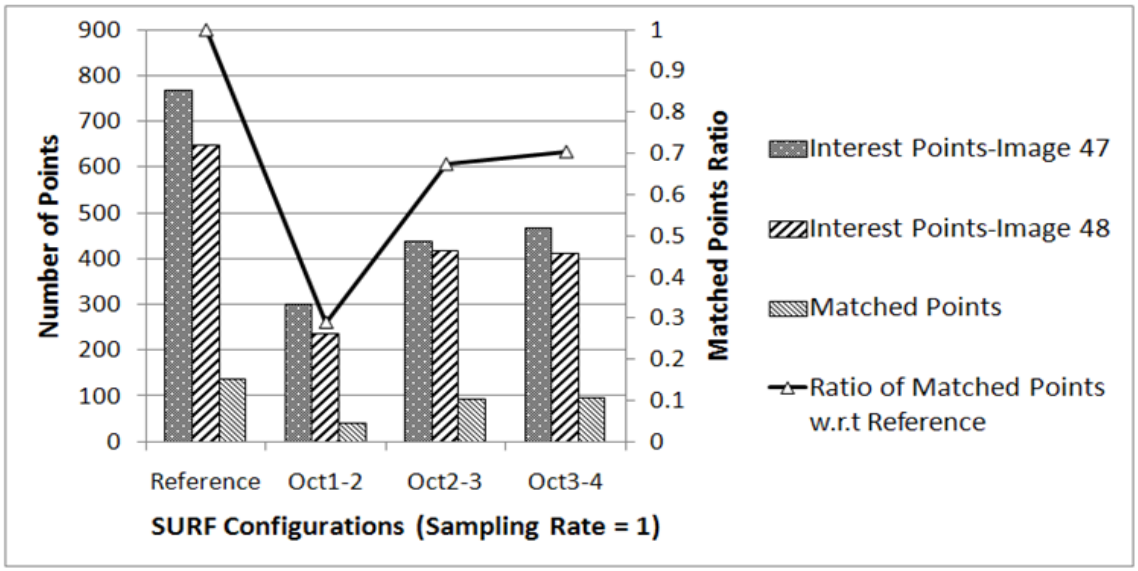

Figure 6-7: Results of best octaves for 47th and 48th image of aerial sequence; octaves 3 and 4 are selected as the best octaves

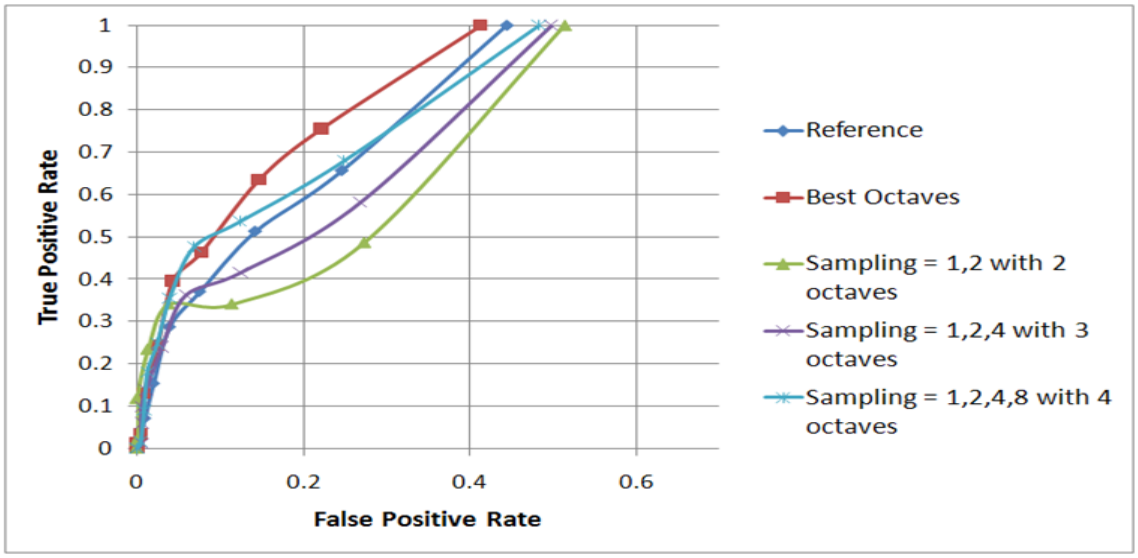

Figure 6-8: Comparison of ROC curves for the 47th and the 48th image of aerial sequence

Figure 6-9: The first (left) and the fifth image (right) of the Trees data set [50]

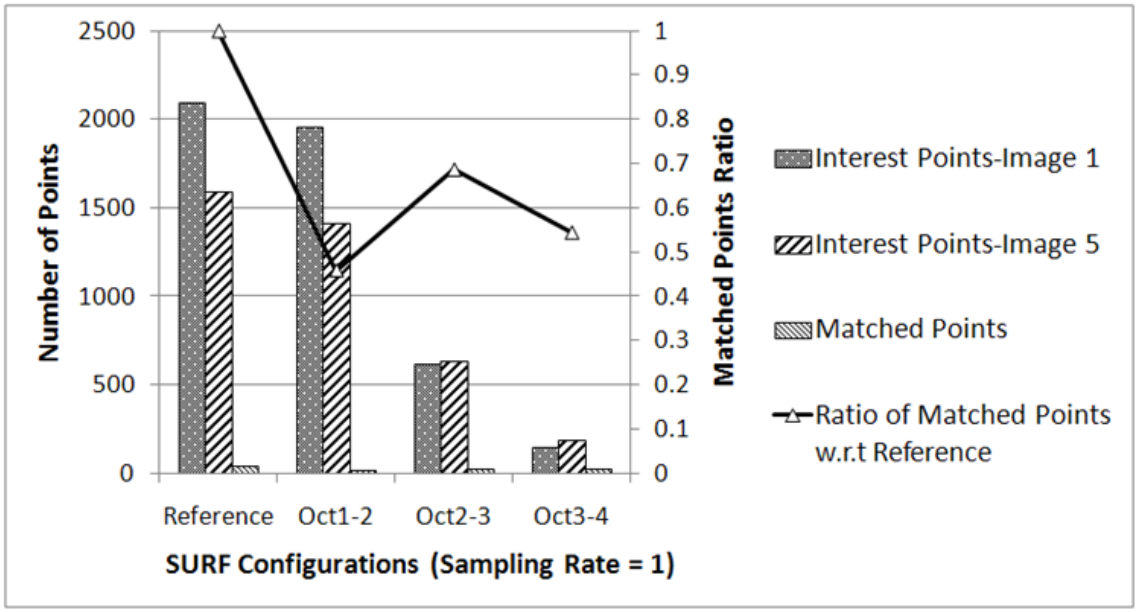

Figure 6-10: Results of best octaves for image 1 and 5 of the Trees data set; octaves 2 and 3 are selected as the best octaves

Finally, results are shown for the first and the sixth images of the widely-used UBC dataset [50] in Figure 6-12. The bars in Figure 6-12 represent the number of points (read values from the left ordinate axis) whereas the line graph shows the matching ratio (read values from the right ordinate axis). Octaves 2 and 3 are the best for UBC images with a matching performance 32.3% less than the reference configuration. Figure 6-13 shows the Sensitivity-Specificity curves for the algorithms, confirming the better performance of the selected best octaves in terms of the quality of interest point matches as compared to the other variants. The interest point

matches obtained using the selected best octaves for image 1 and 6 of the UBC dataset are depicted in Figure 6-14.

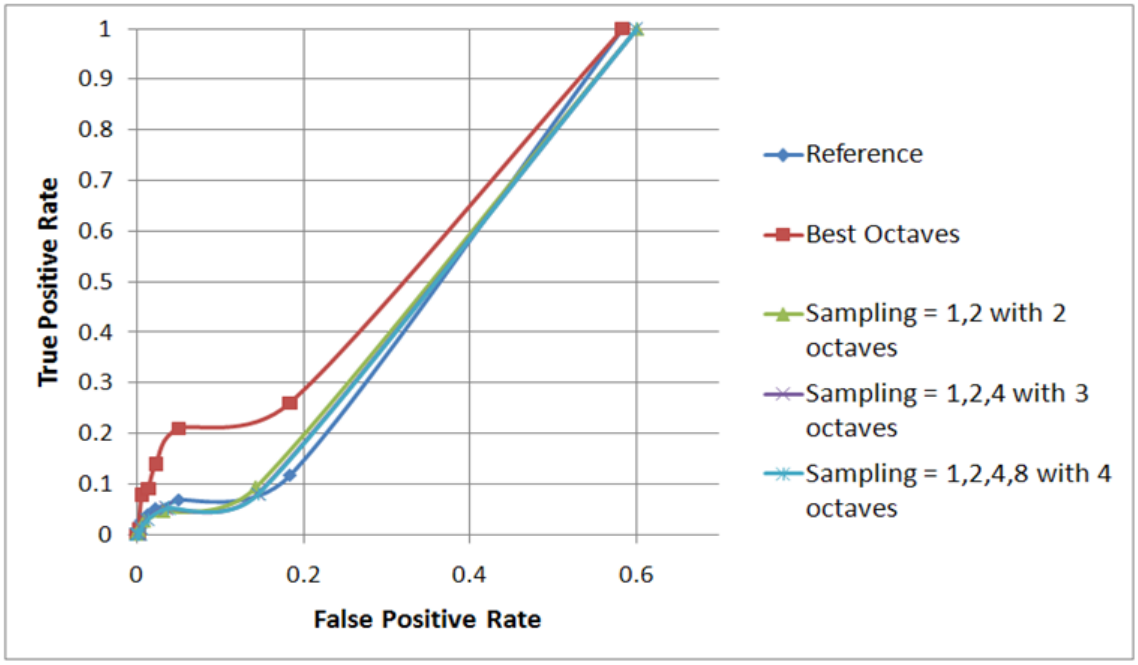

Figure 6-11: Comparison of ROC curves for the first and the fifth image of the Trees data set [50]

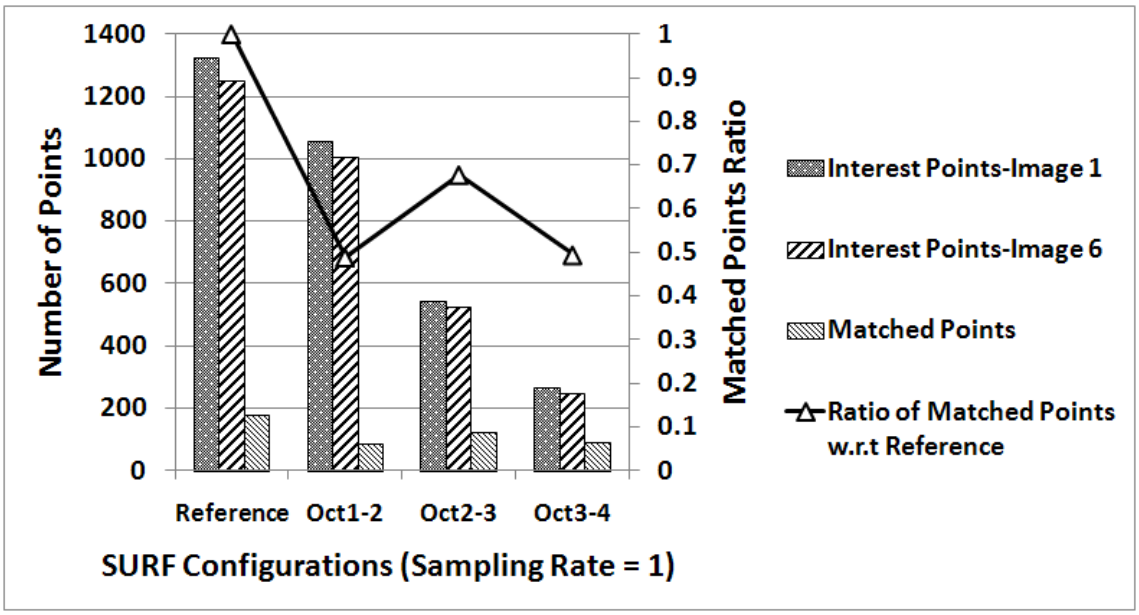

Figure 6-12: Results of best octaves for image 1 and 6 of the UBC data set; octaves 2 and 3 are selected as the best octaves

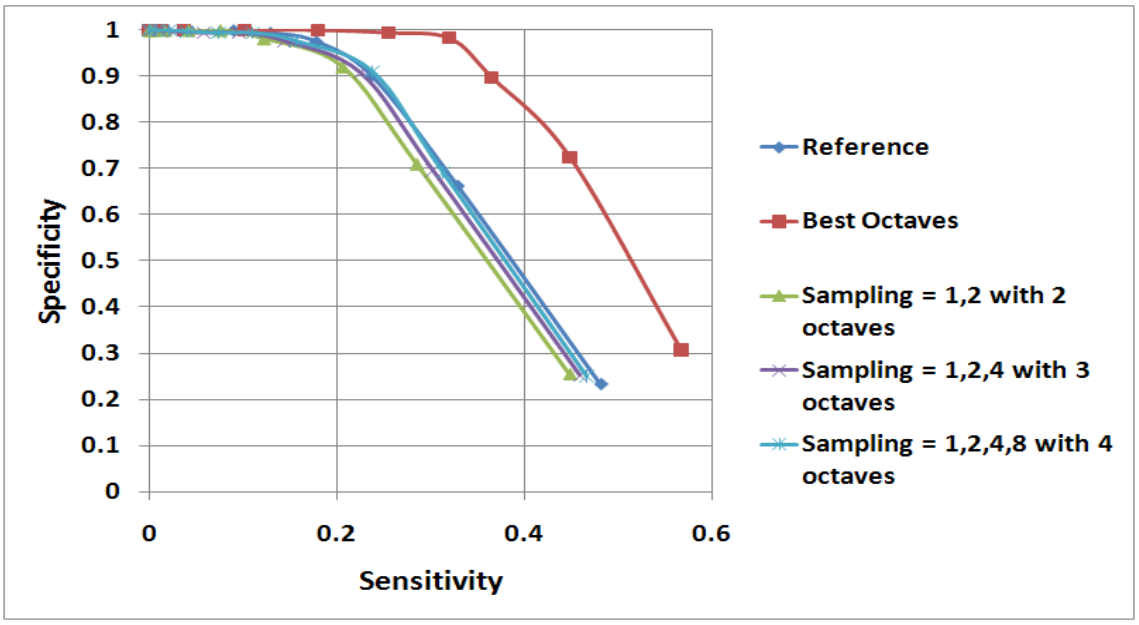

Figure 6-13: Sensitivity-Specificity curves for the first and the sixth images of the UBC dataset

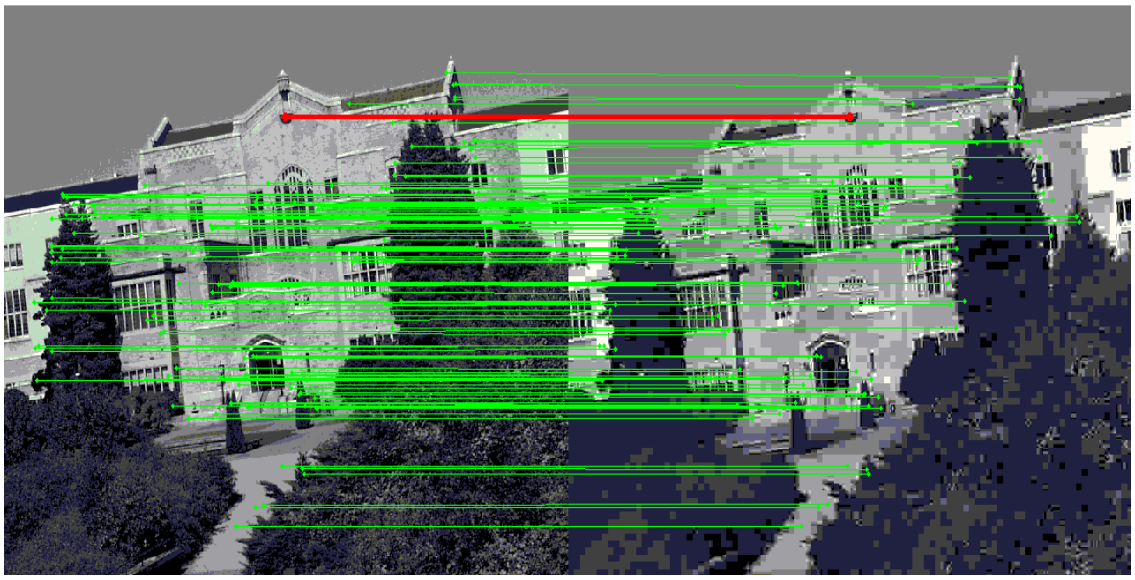

Figure 6-14: Interest point matches obtained using the selected best octaves for the first and the sixth images of the UBC dataset

6.4.3.2 Reduction in Computation

Figure 6-15 provides a comparison in terms of computation of the *best octaves* approach with the maximum performance configuration for the first three stages of SURF. Reductions in computation achieved by *best octaves* for two possible cases are shown: sampling rate of unity and sampling rate of 1, 2 (when octaves 1 and 2 are selected as best octaves). For stage S1, the

computation is equal for the maximum performance configuration and *best octaves*, so there is no reduction in computation, as shown in Figure 6-15. There is however a significant reduction in computation for *best octaves* with respect to the reference configuration in stages S2 and S3. It should also be noted that this reduction in computation is again independent of the specific images being matched.

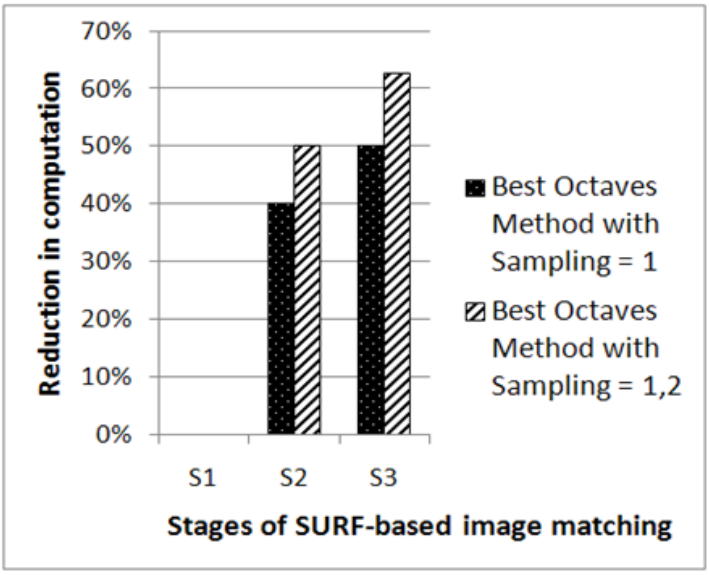

Figure 6-15: Reduction in computation for the first three stages of SURF using best octaves, compared to the maximum performance configuration

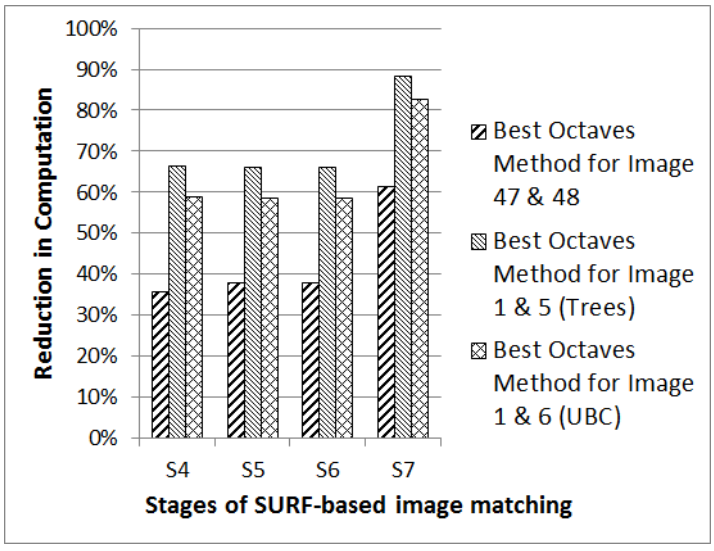

Figure 6-16: Reduction in computation for the last four stages of SURF using best octaves with respect to the maximum performance configuration

A comparison of the computation of *best octaves* with the maximum performance configuration for the last four stages of SURF is shown in Figure 6-16 for the three sample image sets. It is apparent that there is a significant reduction in computation with respect to the maximum performance configuration, demonstrating that the proposed method of octave selection provides a fair trade-off between matching performance and computation with respect to the maximum performance configuration.

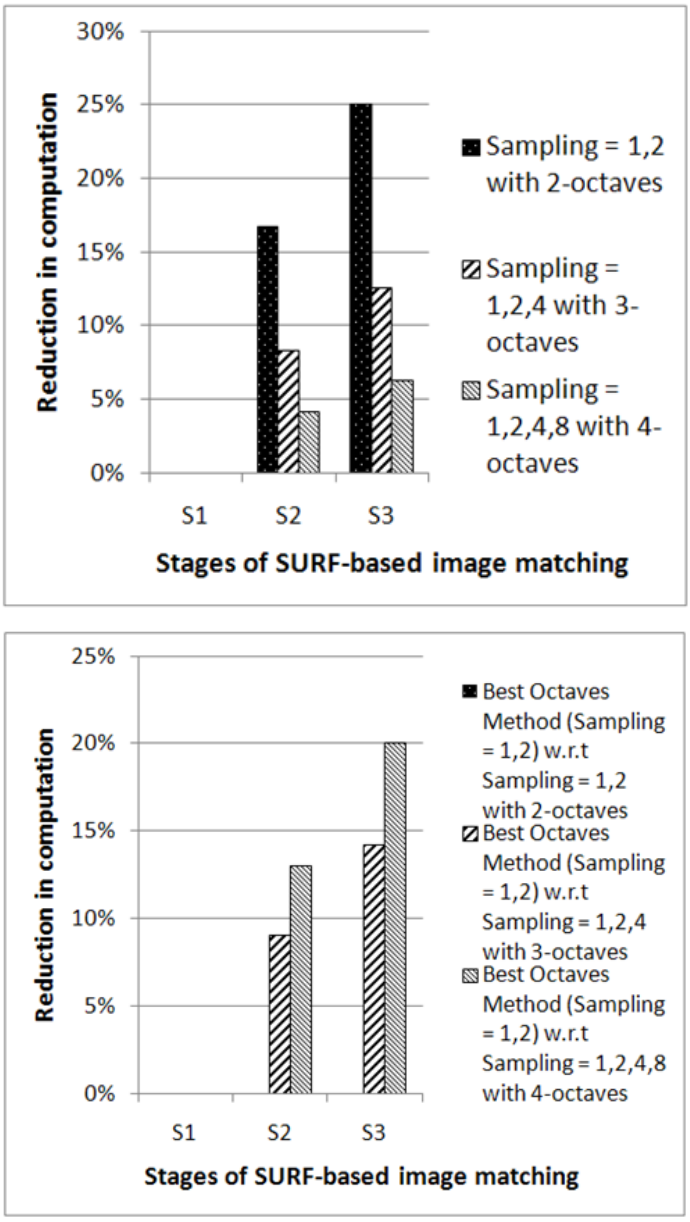

Figure 6-17: Reduction in computation for non-uniformly sampled SURF configurations with respect to best octaves (sampling = 1) for stages S1, S2 and S3 (top) (b) Reduction in computation for best octaves (sampling = 1, 2) with respect to non-uniformly sampled SURF configurations for stages S1, S2 and S3 (bottom)

Figure 6-17 (top) shows the reduction in computation achieved by the three non-uniformly sampled SURF configurations for the first three stages of the algorithm when compared with *best octaves*. This is the case when a sampling rate of unity is selected for *best octaves*. Again, the reduction in computation for the first three stages remains consistent and is independent of the images being matched. Equal computation in stage S1 for all configurations results in no reduction in computation; moreover, it can be seen that the reductions achieved by the non-uniformly sampled SURF configurations with 3 and 4 octaves respectively are not significant.

The reductions for non-uniformly sampled SURF variants in Figure 6-17 (top) are applicable for all situations except when octaves 1 and 2 are selected as the best octaves with non-uniform sampling rate. This particular case is shown in Figure 6-17 (bottom). As can be seen from the zero reductions, best octaves and non-uniformly sampled SURF with 2 octaves require the same amount of computation and provide similar matching performances. It should be noted that when best octaves utilizes non-uniform sampling, it achieves a reduction in computation for stages S2 and S3 with respect to non-uniformly sampled SURF variants with 3 and 4 octaves respectively, whereas there is no reduction in computation for stage S1.

For the three image sets, the percentage reduction in computation for the last four stages of SURF achieved by the best octaves method with respect to non-uniformly sampled SURF configurations having matching performance equal to the best octaves method is presented in Figure 6-18 to Figure 6-20. It is evident that the best octaves approach compensates for the extra computation in the first three stages of the algorithm by achieving significant reduction in computation in the last four stages and comprehensively out-performs the non-uniformly sampled configurations.

Finally, as an example, the author presents here the timings obtained on a mobile robotic platform based on the Intel Atom (CPU N450 running at 1.66 GHz) for extraction of image features and nearest neighbor matching for UBC (image 1 and 6): 30.58, 26.79 and 26.55 sec for non-uniformly

sampled SURF variants with 2, 3 and 4 octaves respectively; and 15.81 sec for best octaves. Clearly, the best octaves approach out-performs the others.

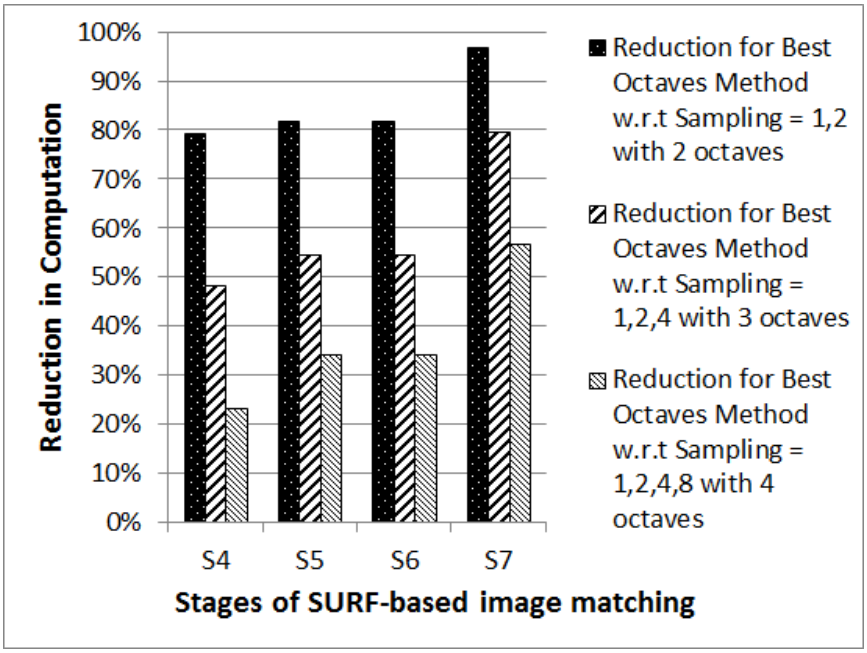

Figure 6-18: Reduction in computation for stages S4, S5, S6 and S7 for best octaves with respect to the non-uniformly sampled SURF configurations having equal matching performance for 47th and 48th image of aerial sequence

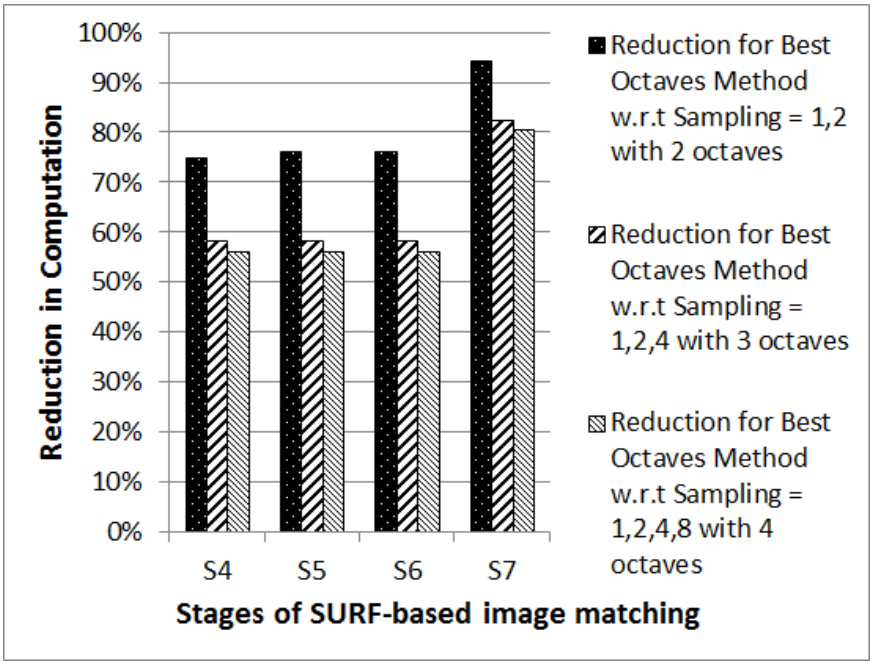

Figure 6-19: Reduction in computation for stages S4, S5, S6 and S7 for best octaves with respect to the non-uniformly sampled SURF configurations having equal matching performance for image 1 and 5 of Trees data set

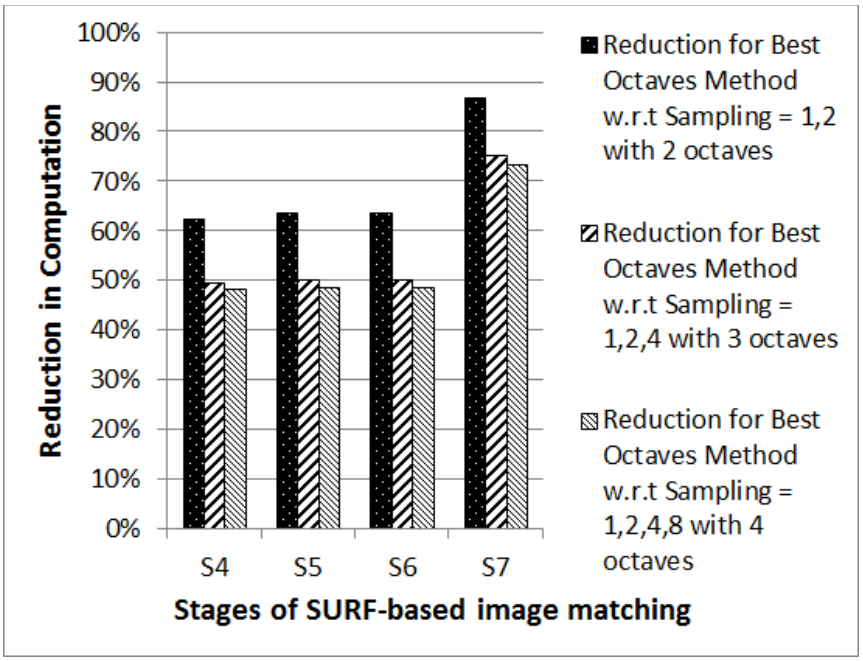

Figure 6-20: Reduction in computation for stages S4, S5, S6 and S7 for best octaves with respect to the non-uniformly sampled SURF configurations having equal matching performance for image 1 and 6 of UBC data set

The results presented illustrate the dominance of the proposed method over non-uniformly sampled SURF configurations in terms of matching performance and computation. To summarize, best octaves provides more and better quality interest point matches for significantly less computation for the three sample cases discussed.

6.5 Statistical Performance Comparison

To back up the largely qualitative discussion of performance in the previous section, it is desirable to be able to perform statistical tests that ascertain whether any differences in performances between *best octaves* and non-uniformly sampled SURF are statistically significant. Formally, one proposes a null hypothesis (*i.e.,* that there is no difference in performance between methods) and uses a statistical test to determine whether the data are consistent with this hypothesis. As already mentioned in Section 5.3.3, the appropriate statistic in this case is McNemar's test, a form of chi-squared test that evaluates the performance of the two algorithms based on

their outcomes on a case-by-case basis over the same dataset [51, 52] (see Equation 4-6).

McNemar's test was used to compare the performance of the *best octaves* method with non-uniformly sampled SURF systems with each of 2, 3 and 4 octaves. To employ it, one needs a criterion for determining whether a particular case results in success or failure; as a matched interest point is more significant than a detected interest point in vision applications, the criterion adopted is based on the number of matches obtained. Since the maximum performance configuration (sampling rate of unity with 4 octaves) provides the maximum number of matches, if an algorithm achieves at least 45% of the number obtained by this maximum performance configuration, it is considered to have succeeded; otherwise, it is deemed to have failed. Although the 45% figure is arbitrary, it does not affect the conclusions we draw from the data. To avoid inadvertent dataset dependencies, a total of 776 image pairs were employed, taken from the Oxford [13], Copydays [213] and Blur [214] datasets used in [46, 55, 215, 216]. The image pairs involved changes in scale, rotation, blurring, illumination, viewpoint and JPEG compression. All results were obtained using OpenSURF at a threshold of 0.0022.

6.5.1 Matching Performance

Table 6-6 shows the results for *best octaves* and non-uniformly sampled SURF with two octaves. There are 358 image pairs for which the non-uniformly sampled SURF with 2 octaves failed to achieve 45% of the maximum matching performance, cases where *best octaves* succeeded. More significantly, there is not a single image pair for which *best octaves* failed and 2 octaves SURF passed. The resulting *Z-score* for these results is 18.8, indicating that *best octaves* out-performs 2-octaves SURF (sampling = 1, 2) with a probability well in excess of 99.5%.

Table 6-6: Results of McNemar's Test for best octaves and non-uniformly sampled SURF with 2-octaves

	2-octaves SURF (Sampling = 1, 2) PASS	**2-octaves SURF (Sampling = 1, 2) FAIL**
Best Octaves PASS	402	358
Best Octaves FAIL	0	16

Results comparing *best octaves* and non-uniformly sampled SURF with 3 octaves are presented in Table 6-7. The non-uniformly sampled 3-octaves SURF performs better than its 2-octave variant with a probability well in excess of 99.5%. However, there are still 168 image pairs for which 3-octaves SURF failed to achieve the 45% correct matches threshold when *best octaves* succeeded; yet there are only 16 image pairs for which the converse occurred. Using Equation 4-6, the value of Z is 11.1, showing that *best octaves* comprehensively outperforms 3-octaves SURF (sampling = 1, 2) variant with a probability well in excess of 99.5%.

Table 6-7: Results of McNemar's test for best octaves and non-uniformly sampled SURF with 3-octaves

	3-octavesSURF (Sampling = 1, 2, 4) PASS	**3-octavesSURF (Sampling = 1, 2, 4) FAIL**
Best Octaves PASS	592	168
Best Octaves FAIL	16	0

Finally, *best octaves* was compared statistically with 4-octaves SURF (sampling = 1, 2, 4, 8) using McNemar's test (Table 6-8). Equal numbers of

failures of both algorithms were obtained, resulting in a *Z-score* of zero and showing that the performances of the two algorithms are statistically indistinguishable. However, *best octaves* required only 2 octaves to achieve this, whereas the non-uniformly sampled SURF needed to compute 4 octaves.

Table 6-8: Results of McNemar's test for best octaves and non-uniformly sampled SURF with 4-octaves

	4-octaves SURF (Sampling = 1, 2, 4, 8) PASS	**4-octaves SURF (Sampling = 1, 2, 4, 8) FAIL**
Best Octaves PASS	744	16
Best Octaves FAIL	16	0

6.5.2 Reduction in Computation

Since the reduction in computation achieved by the non-uniformly sampled SURF variants with respect to *best octaves* for the first three stages of the algorithm is independent of images being analyzed, it is interesting to examine the performance of *best octaves* in terms of computation for the last four stages (S4–S7) of the algorithm. To make this analysis thorough, the amount of computation required by every algorithm in the last four stages for the 776 image pairs used in McNemar's test above was measured. Since it is tedious to equate the matching performance of all the algorithms by varying their threshold values (as was done in Table 6-5) for such a large number of image pairs, the difference in the number of interest points processed per match is used for comparing the computation of algorithms. Since computation in stages S4, S5 and S6 of SURF is a function of the number of detected local maxima and interest points (Table 6-1), this allows a fair comparison between *best octaves* and the non-uniformly sampled

SURF variants for these stages. For a comparative analysis of computation for the last stage, interest point matching (S7), the difference in the number of descriptor comparisons per match is used.

For every image pair, the number of interest points processed per match is calculated for *best octaves* and the 2-octaves SURF (sampling = 1, 2). The values obtained for *best octaves* are subtracted from the number of interest points processed per match for the 2-octaves SURF configuration so that a positive value indicates that the best octaves method does less computation per match. To gauge the significance and magnitude of any computation reduction achieved by *best octaves*, Figure 6-21 shows the histogram of the difference in number of interest points processed per match (*i.e.,* for stages S4–S6). It is evident that there are only 9 instances where computation of *best octaves* is higher than 2-octaves SURF. Indeed, the mean difference in the number of interest points processed per match in this particular case is 44.7, showing that *best octaves* computes and processes *nearly 45 times* fewer interest points than non-uniformly sampled SURF with 2-octaves, a huge reduction in computation. The histogram of the difference in number of descriptor comparisons per match (stage S7) is shown in Figure 6-22 for this particular case, again illustrating the significant reduction in computation achieved by *best octaves*. On average, 2-octaves SURF (sampling = 1, 2) performs 14,940 more descriptor comparisons per match than *best octaves*.

Figure 6-23 shows the histogram of differences in the number of interest points processed per match for *best octaves* and 3-octaves SURF (sampling = 1, 2, 4). The distribution indicates that 3-octaves SURF achieves a reduction in computation with respect to *best octaves* for only 190 image pairs. Although the performance of 3-octaves SURF is better than 2-octaves, it still computes 12 times more interest points than *best octaves* on average. Thus, *best octaves* achieves a significant reduction in computation here as well. To illustrate the reduction in computation achieved for the interest point matching stage, a histogram of the differences in the number of descriptor comparisons per match is shown in Figure 6-24. *Best octaves* is

again dominant as the non-uniformly sampled 3-octaves SURF requires on average 6,159.5 more descriptor comparisons per match.

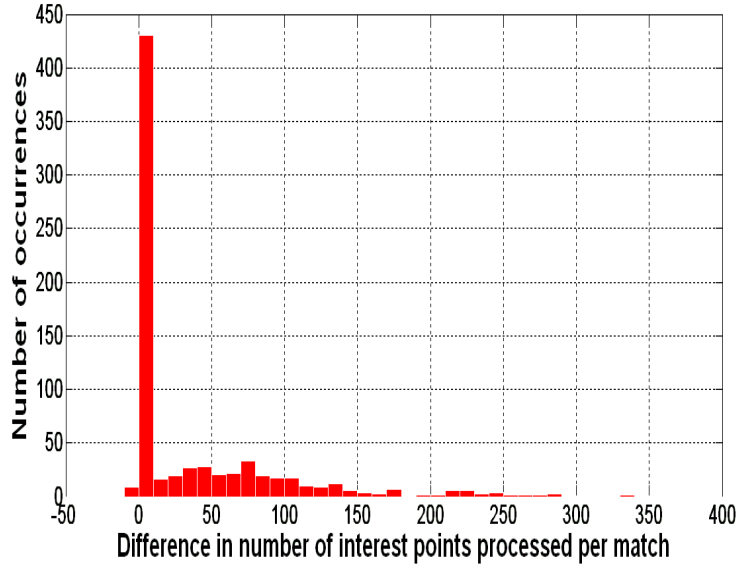

Figure 6-21: Histogram of difference in number of interest points processed per match for best octaves and non-uniformly sampled SURF with 2-octaves

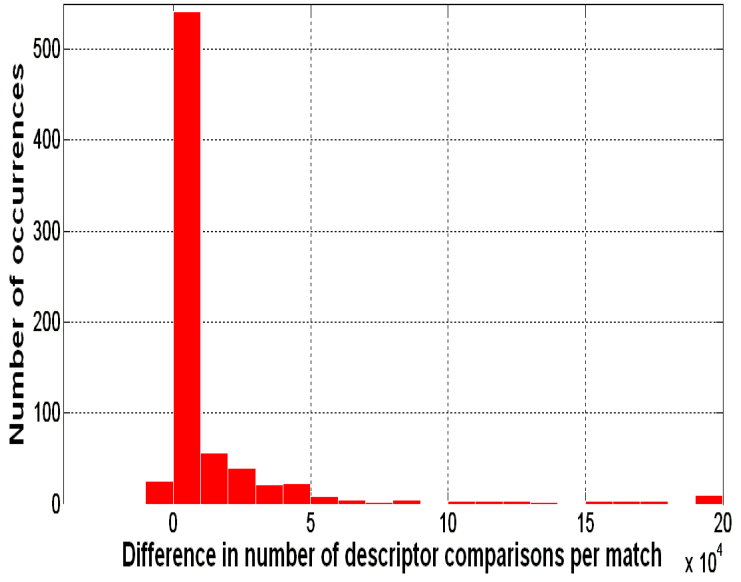

Figure 6-22: Histogram of difference in number of descriptor comparisons per match for best octaves and non-uniformly sampled SURF with 2-octaves

Finally, a comparison of computation for *best octaves* and the non-uniformly sampled SURF configuration with 4-octaves is presented in Figure 6-25 and Figure 6-26. The two histograms again demonstrate the dominance of *best octaves:* on average, it processes 5 times fewer interest points than 4-octaves SURF. Similarly, for the matching stage, *best octaves* achieves a significant reduction in computation as 4-octaves SURF requires 3,991.4 more descriptor comparisons per match on average.

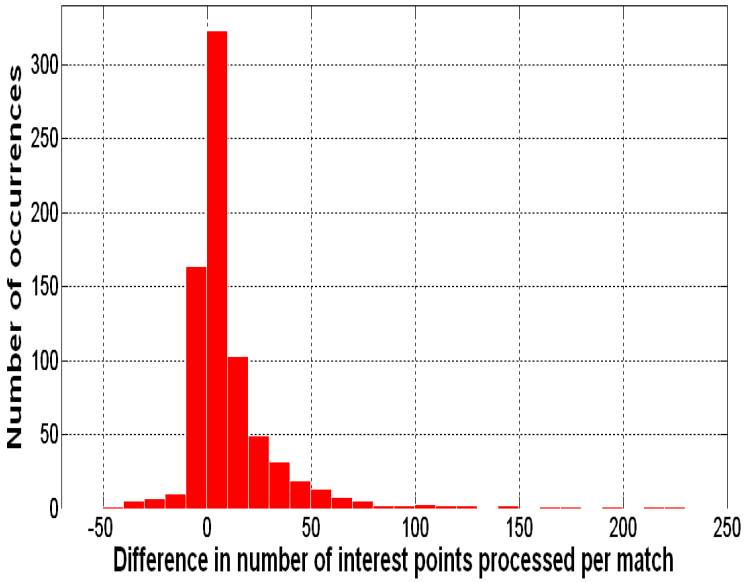

Figure 6-23: Histogram of difference in number of interest points processed per match for best octaves and non-uniformly sampled SURF with 3-octaves

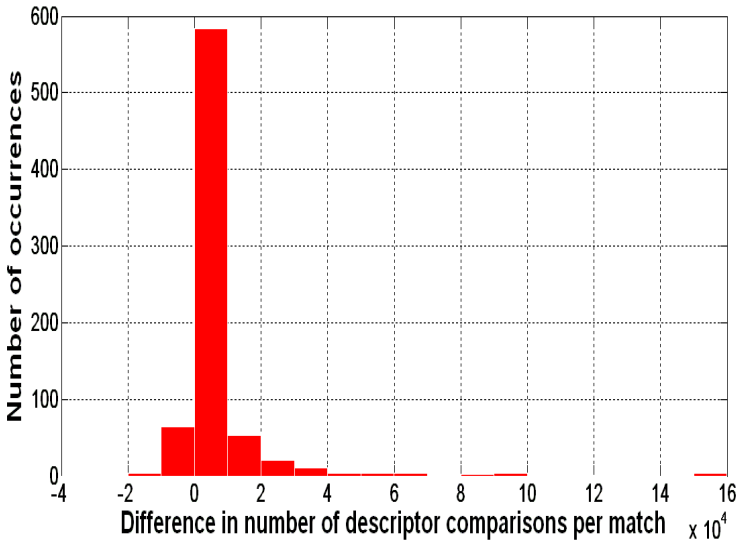

Figure 6-24: Histogram of difference in number of descriptor comparisons per match for best octaves and non-uniformly sampled SURF with 3-octaves

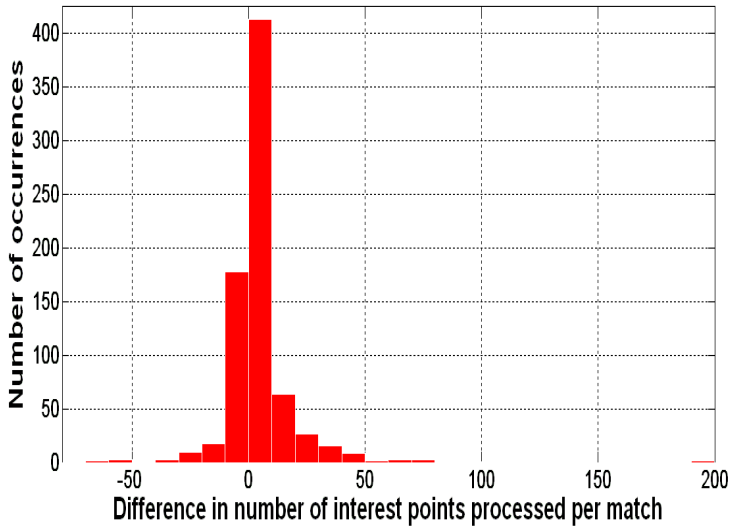

Figure 6-25: Histogram of difference in number of interest points processed per match for best octaves and non-uniformly sampled SURF with 4-octaves

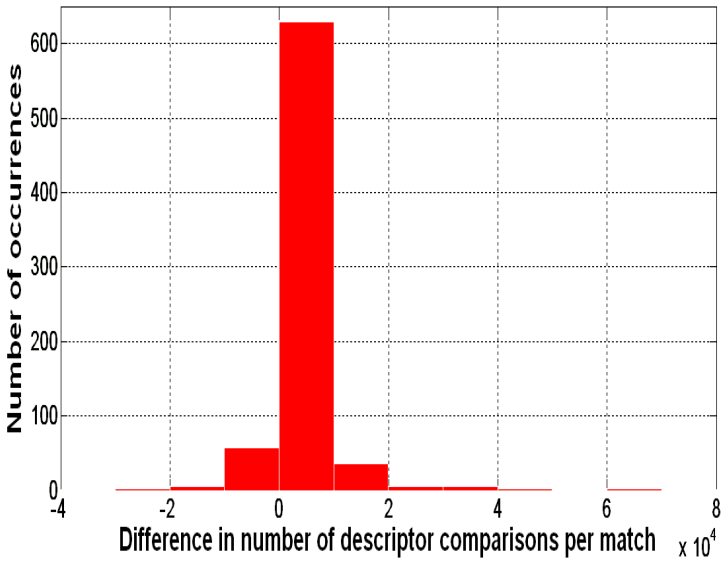

Figure 6-26: Histogram of difference in number of descriptor comparisons per match for best octaves and non-uniformly sampled SURF with 4-octaves

6.6 Summary

Algorithm-level optimization of SURF is critical for designing highly-optimized software and/or hardware implementations of the algorithm – and doing so is essential if real-time performance is to be achieved on commodity hardware. After drawing attention to the limitations of the conventional method of octave reduction for enhancing execution speed, this work has underlined the significance of employing a unity sampling rate for the

detection stages of SURF, providing an equal opportunity for all octaves to achieve their maximum performance. The chapter has proposed a method for reducing the computational complexity of SURF, namely an intelligent reduction in the number of SURF octaves. As opposed to the conventional method that concentrates only on reducing the computational complexity of detection stages, the proposed algorithm sets a new paradigm by emphasizing on the description and matching stages of SURF to yield significant reduction in computation at the cost of little extra calculation in the detection stages. Both software and hardware solutions can benefit from the proposed method: for example, on multi-core processors, a multi-threaded implementation of SURF can potentially achieve more speed-up using the presented algorithm. In addition, this work may pave the way for the use of a computation-intensive technique like SURF in battery-operated robots, for which low-power consumption is extremely critical. It has been shown that, as well as reducing the computation significantly, this approach also out-performs other SURF variants in terms of matching performance.

7 Integral Images: Efficient Algorithms for their Computation and Storage

Innovation distinguishes between a leader and a follower.

STEVE JOBS

The *integral image* is an intermediate image representation that allows rapid calculation of rectangular features at constant speed, irrespective of filter size, and is particularly useful for multi-scale local feature detection algorithms like Speeded-Up Robust Features (SURF). Although calculation of the integral image involves simple addition operations, the total number of operations is significant due to the generally large size of image data. Recursive equations allow considerable reduction in the required number of addition operations but require calculation of the integral image in a serial fashion. This is generally not desirable for real-time embedded vision systems with strict time limitations and low-powered but parallel hardware resources. With the objective of minimizing the computational resources involved, this chapter proposes two novel hardware algorithms based on the decomposition of these recursive equations, allowing calculation of up to four integral image values in a row-parallel way without significantly increasing the number of addition operations. An efficient design strategy is also proposed for a parallel integral image computation unit to reduce the size of the required internal memory. Finally, two algorithms which allow substantial decrease in the memory requirements for the storage of the integral image are presented.

7.1 Introduction

Originally proposed as the *summed-area table* for texture-mapping in computer graphics in the mid-1980s [217], the integral image is comparatively new in the image processing domain. The idea of using an integral image was introduced as an intermediate image representation by the Viola-Jones face detector [132]. Since then, it has been particularly useful for fast implementation of image pyramids in multi-scale computer vision algorithms such as Speeded-Up Robust Features (SURF) and Fast Approximated SIFT [13, 53, 218].

The primary reason for using an integral image is the improved execution speed for computing box filters. Employment of the integral image eliminates computationally expensive multiplications for box filter calculation, reducing it to three addition operations [132]. This allows all box filters to be computed at a constant speed, irrespective of their size; this is a major advantage for computer vision algorithms, especially feature detection techniques which utilize multi-scale analysis. Such algorithms generally require calculation of variable-size box filters to implement different scales of an image pyramid. For example, SURF requires computation of 9 x 9 box filters for implementation of the smallest and 195 x 195 for the largest scale of its image pyramid [13]; without an integral image, these larger filters would take almost 500 times longer than the smallest one to compute.

Although speed gain and reduced computational complexity are major benefits of the utilization of the integral image representation, the calculation of integral image introduces a performance overhead [219]. Image processing and computer vision algorithms are generally computation and data intensive in nature, and integral image calculation is no exception. Although it involves only additions, the total number of operations is significant due to its dependence upon the input image size. Recursive equations due to Viola and Jones [132] reduce the total number of additions required for computation of the integral image but require that calculation

is done in a serial fashion because of the data dependencies involved. This is not desirable for real-time embedded vision systems that have strict time limits and restricted hardware resources for processing a single frame, possibly coupled with power constraints.

Since serial calculation can provide only one integral image value per clock cycle at best, there is a strong motivation to investigate methods for efficient computation of the integral image. Indeed, there are examples in the literature where efficient computation of the integral image has been achieved on a variety of computing platforms such as multi-core processors, GPUs (Graphics Processing Units), and custom hardware [219-242]. For example, integral image calculation is accelerated by first computing the sum of all pixels in the horizontal direction and then in the vertical direction utilizing the huge computational resources of a GPU (ATI HD4850 in this particular case) in [220]. However, to the author's knowledge, no technique has emerged so far that would achieve significant speed-up for integral image computation while optimizing the required computational and memory resources which is a big constraint for embedded systems. This chapter takes a step in this direction. Firstly, it performs an analysis of the recursive equations and the data dependencies involved for parallel calculation of integral image; it then proposes two hardware algorithms based on the decomposition of these recursive equations, allowing simultaneous computation of up to four integral image values in a row-parallel way without any significant increase in the number of addition operations. An efficient design strategy for a parallel integral image computation engine is then presented which reduces the internal memory requirements significantly.

Another drawback of the utilization of the integral image representation is the substantial increase in the memory requirements for its storage [243]. This is essentially due to the significantly larger word length of integral image values as compared to the original pixel values. Again, for embedded vision systems it becomes a bottleneck due to the strict constraints on hardware resources. In [243], two techniques are presented

for reducing the word length of integral image: an exact method which reduces the word length by computation through the overflow without any loss of accuracy on platforms with complement-coded arithmetic; and an approximate technique which is based on rounding the input image by value truncation. The exact method is useful only when the maximum size of the box filter is considerably smaller than the input image size. Loss of accuracy is the main drawback of the approximate method. To address these issues, this chapter presents two generic methods for reducing the storage requirements of the integral image significantly which can benefit both custom hardware design and software implementation on programmable processor architectures for resource constrained embedded vision systems.

The remainder of this chapter is structured as follows. An analysis of the computation of the integral image is given in Section 7.2. Proposed in Section 7.3 is a parallel computation strategy that provides two integral image values per clock cycle. Section 7.4 presents another parallel method that delivers four integral image values per clock cycle. Extending the approach of Section 7.3, a memory-efficient design strategy is proposed for a parallel integral image computation unit in Section 7.5. Two methods for reducing the size of memory for storing integral image are presented in Section 7.6. Finally, a summary of the chapter is provided in Section 7.7.

7.2 Analysis of Integral Image Computation

This section analyzes integral image calculation from a parallel computation perspective. The value of the integral image at any location (x,y) in an image is the sum of all the pixels to the left of it and above it, including itself, as shown in Figure 7-1. This can be stated mathematically as in [132]:

$$ii(x,y) = \sum_{x' \le x,\, y' \le y} i(x',y') \qquad \textit{Equation 7-1}$$

where $ii(x,y)$ and $i(x,y)$ are the values of the integral image and the input image respectively at location (x,y).

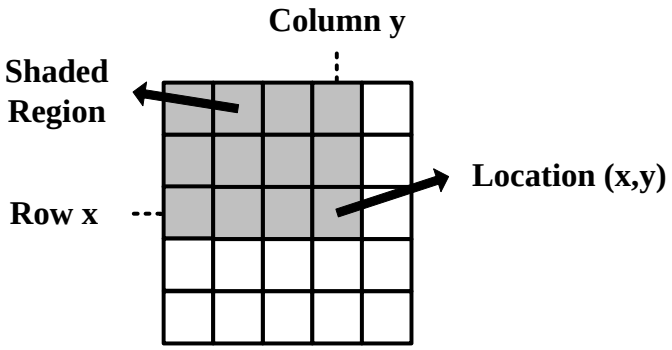

Figure 7-1: Calculation of integral image value at image location (x,y). The shaded region indicates all pixels to be summed

Equation 7-1 has potential for parallel computation, providing the input image is stored in memory and all its pixel values can be accessed. For example, the integral image of a 2 x 2 image may be computed in parallel using the following set of equations:

$$ii(1,1) = i(1,1) \qquad \textit{Equation 7-2}$$

$$ii(1,2) = i(1,1) + i(1,2) \qquad \textit{Equation 7-3}$$

$$ii(2,1) = i(1,1) + i(2,1) \qquad \textit{Equation 7-4}$$

$$ii(2,2) = i(1,1) + i(1,2) + i(2,1) + i(2,2) \qquad \textit{Equation 7-5}$$

Although Equation 7-1 can be used for small images, the number of additions involved scales as $\frac{1}{4}M^2N^2$for an input image of size M x N pixels [219]. For example, 1,866,240,000 addition operations are required to compute the integral image for a medium resolution image of size 360 x 240 pixels. Thus, Equation 7-1 is not particularly suitable from a hardware perspective.

The total number of addition operations can be drastically reduced by utilizing the recursive equations presented in [132]:

$$S(x,y) = i(x,y) + S(x,y-1) \qquad \textit{Equation 7-6}$$

$$ii(x,y) = ii(x-1,y) + S(x,y) \qquad \textit{Equation 7-7}$$

where $i(x,y)$ is the input pixel value at image location (x,y), $S(x,y)$ is the cumulative row sum value at image location (x,y) and $ii(x,y)$ is the integral image value at image location (x,y). These equations reduce the number of additions involved to $2MN$.

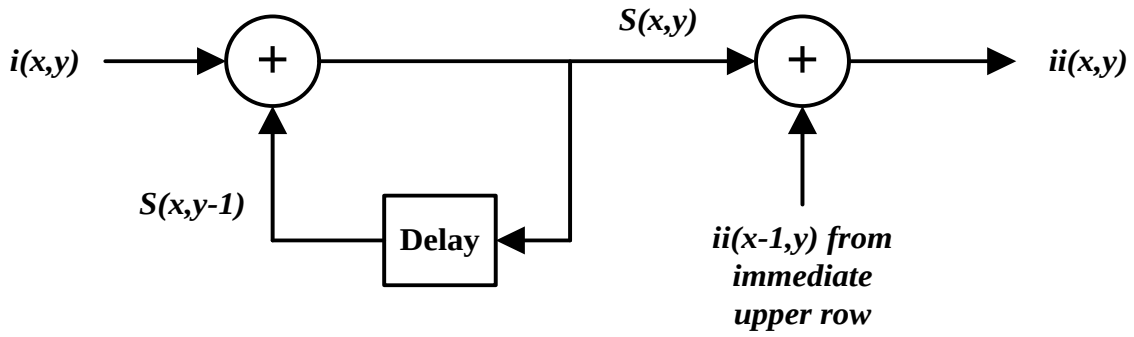

Figure 7-2: Data Flow Graph of the Viola-Jones recursive equations for a single row of the input image

Equation 7-6 and Equation 7-7 represent a two-stage system which operates in a serial fashion: the first stage computes the cumulative row sum at a specific image location and forwards the data to the second stage for calculation of the integral image value at that particular location. The data flow graph of this serial system is shown in Figure 7-2 for a single row of the input image. It can be observed from Figure 7-2 that individual stages are also dependent upon data from previous iterations for their operation. The first stage requires the cumulative row sum to be computed in a serial way for a single row of the input image. The second stage is more complex as it needs data from the previous row to calculate an integral image value. Hence, there is little opportunity for parallel computation in single row operations.

However, a deeper analysis of Equation 7-6 and Equation 7-7 shows that it is possible to compute the cumulative row sum for all rows independently and hence simultaneously. This is however not true for Equation 7-7 due to its dependency on data from the neighboring row.

Thus, the best possible system using these equations is to process individual rows in a delayed fashion. As an example, Figure 7-3 shows a 5 x 5 image for which integral image values are calculated by processing all rows in parallel using these equations. The shaded blocks represent the pixels for which integral image values are computed simultaneously; blocks with a cross sign indicate pixels whose integral image values have already been calculated; and white blocks show pixels for which integral image values still need to be calculated. It can easily be seen that the integral image value for the second pixel in the third row cannot be calculated until the integral image value for the second pixel in the second row is calculated. Figure 7-4 shows the time delay involved in computation of integral image values for different rows.

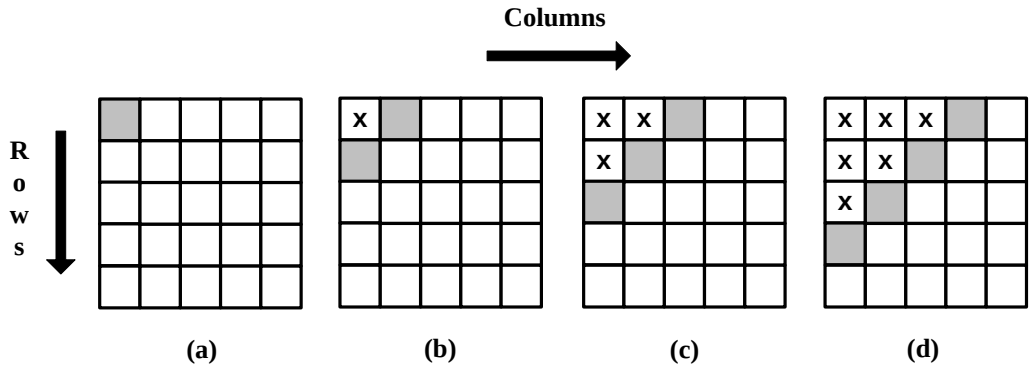

Figure 7-3: Delayed row computation using the Viola-Jones recursive equations

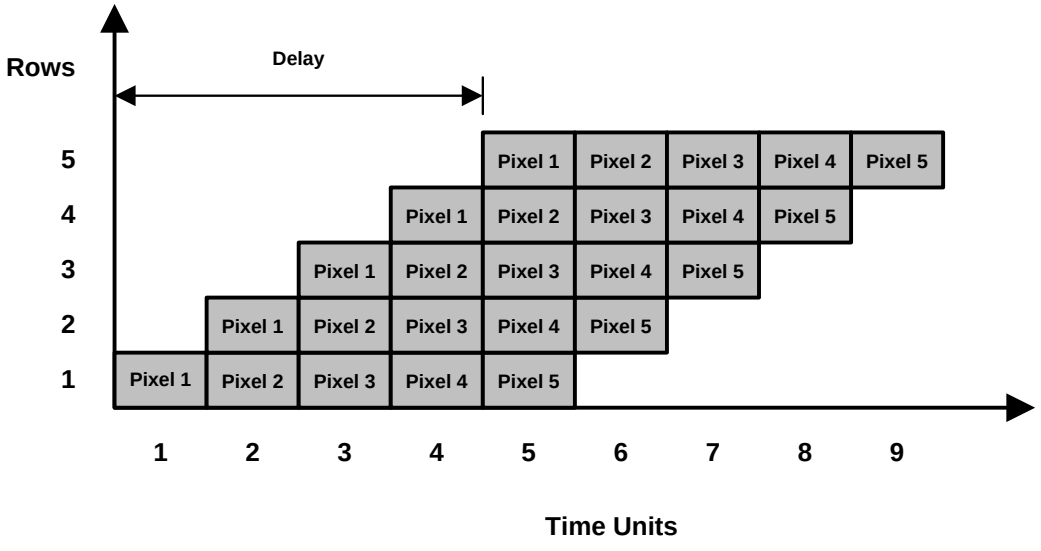

Figure 7-4: Time delay between computation of integral image values for different rows

7.3 Parallel Computation for Two Rows

The proposed algorithm represents a two-stage, pipelined system that processes two rows of an input image in parallel, providing two integral image values per clock cycle without any delay when the pipeline is full. In particular, it allows calculation of the second pixel of the two rows in the *same* clock cycle. The whole image is divided in groups of two rows and one group is processed at a time, moving from the top to the bottom of the input image. The following set of equations is used for calculation of integral image values in a row-parallel way:

$$S(x,y) = i(x,y) + S(x,y-1) \qquad \textit{Equation 7-8}$$

$$S(x+1,y) = i(x+1,y) + S(x+1,y-1) \qquad \textit{Equation 7-9}$$

$$ii(x,y) = ii(x-1,y) + S(x,y) \qquad \textit{Equation 7-10}$$

$$ii(x+1,y) = ii(x-1,y) + S(x,y) + S(x+1,y) \qquad \textit{Equation 7-11}$$

where Equation 7-8 and Equation 7-10 are for computation of integral image values in the first row; and Equation 7-9 and Equation 7-11 are for the second row.

This set of equations requires $2MN + \frac{MN}{2}$ addition operations for an input image of size M x N pixels. This is not a significant increase compared to the $2MN$ additions required for the standard recursive equations, Equation 7-6 and Equation 7-7. For all odd rows, two additions are required per pixel, as given by Equation 7-8 and Equation 7-10. An extra addition is done for each pixel in the even rows in Equation 7-11 to allow simultaneous calculation of integral image values for even and odd rows without any delay. The data flow graph for the proposed algorithm is shown in Figure 7-5. A pipelined approach for this two-stage system reduces the critical data path from two adders to one. The proposed system has the potential to compute integral images in half the number of clock cycles required serially.

For example, for an input image of size 640 x 480 pixels, 307200 clock cycles are required for conventional serial calculation, whereas this approach requires only 153600.

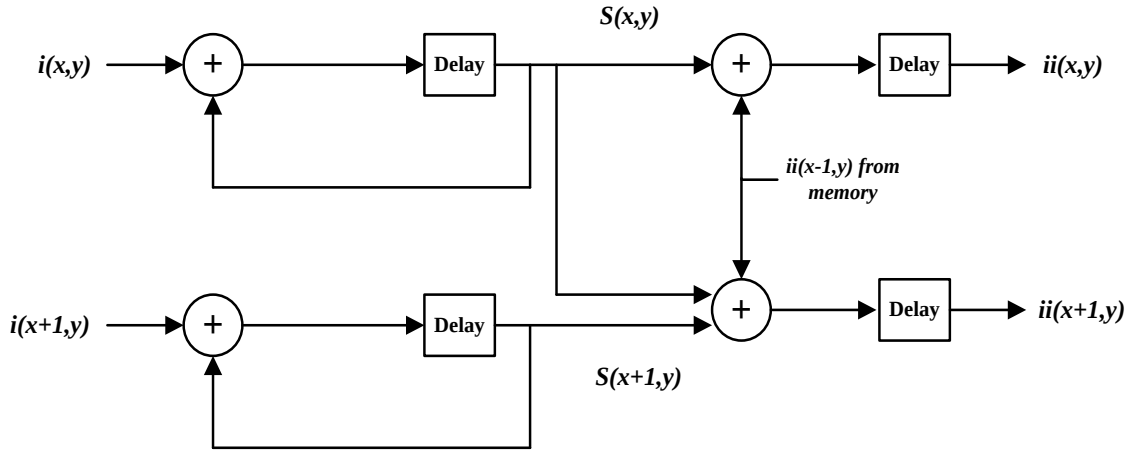

Figure 7-5: Data Flow Graph for Parallel computation of integral image for 2 rows

7.4 Parallel Computation for Four Rows

The above algorithm for processing two rows in parallel can be extended to four rows, though at the expense of extra additions per pixel in rows 3 and 4, allowing calculation of four integral image values per clock cycle. However, this is not an attractive option as it involves more hardware. With the objective of minimizing hardware resources, another decomposition of Equation 7-6 and Equation 7-7 is proposed in this section; it provides four integral image values per clock cycle in a row-parallel way with $2MN + \frac{MN}{2}$ additions for an input image of size M x N pixels.

The proposed algorithm represents a three-stage, pipelined system (as opposed to the two-stage one above) to reduce the computational resources required in hardware. It processes four rows of an input image in parallel, providing *four* integral image values per clock cycle. In this case, the image is divided in groups of four rows and one group is processed at a time moving from top to bottom. The following set of equations is used for calculation of integral image values in a row-parallel way:

$$S(x,y) = i(x,y) + S(x,y-1) \quad \textit{Equation 7-12}$$

$$S(x+1,y) = i(x+1,y) + S(x+1,y-1) \quad \textit{Equation 7-13}$$

$$S(x+2,y) = i(x+2,y) + S(x+2,y-1) \quad \textit{Equation 7-14}$$

$$S(x+3,y) = i(x+3,y) + S(x+3,y-1) \quad \textit{Equation 7-15}$$

$$ii(x,y) = ii(x-1,y) + S(x,y) \quad \textit{Equation 7-16}$$

$$ii(x+1,y) = ii(x-1,y) + S(x,y) + S(x+1,y) \quad \textit{Equation 7-17}$$

$$ii(x+2,y) = ii(x+1,y) + S(x+2,y) \quad \textit{Equation 7-18}$$

$$ii(x+3,y) = ii(x+1,y) + S(x+2,y) + S(x+3,y) \quad \textit{Equation 7-19}$$

where Equation 7-12 and Equation 7-16 are for computation of integral image values in the first row; Equation 7-13 and Equation 7-17 are for the second row; Equation 7-14 and Equation 7-18 are for the third row; and Equation 7-15 and Equation 7-19 are for the fourth row.

The main advantage of this system is that it requires $2MN + \frac{MN}{2}$ addition operations for an input image of size M x N pixels as is required for parallel processing of 2 rows. The corresponding data flow graph is shown in Figure 7-6. This scheme has the potential to compute the integral image for an input image in one-fourth the number of clock cycles required for serial calculation, reducing the number of clock cycles for calculation of the integral image of a 640 x 480 image to only 76800.

Table 7-1 presents comparative resource utilization results for prototype implementations of the serial method (Viola-Jones recursive equations), the proposed two-rows, and the 4-rows algorithms on a Xilinx Virtex-6 FPGA for some common image sizes. All three implementations achieve a maximum frequency of about 147 MHz. It is evident that, without

significant increase in the utilized resources, the proposed algorithms provide two and four times speed-up relative to the serial algorithm. The resource consumption for the three compared algorithms increases with the increasing image size and all the algorithms show a similar trend. This is essentially due to the same internal memory requirements (to store one complete row of integral image values for the calculation of the very next row) of the three algorithms.

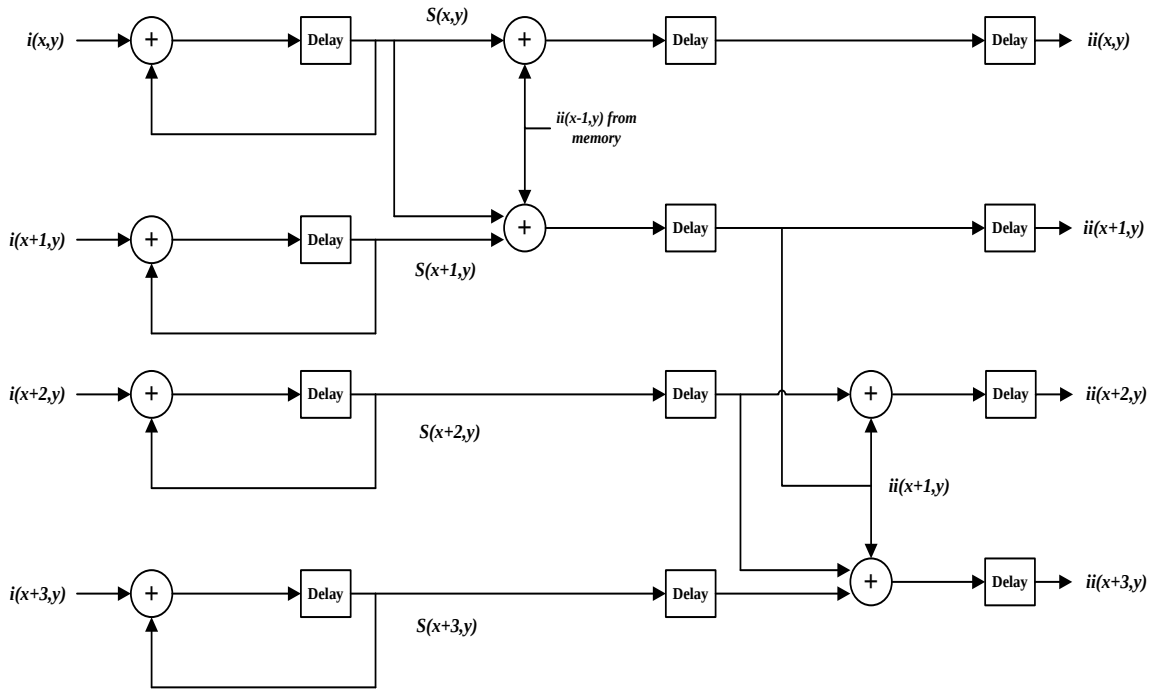

Figure 7-6: Data Flow Graph for Parallel computation of integral image for 4 rows

Table 7-1: Comparative resource utilization results for Serial, 2-rows and 4-rows parallel prototype implementations on a Xilinx Virtex-6 XC6VLX240T FPGA for some common image sizes

Image Size	Serial		2-Rows Parallel		4-Rows Parallel	
	Slice Registers	LUTs	Slice Registers	LUTs	Slice Registers	LUTs
360 x 240	9050	3465	9075	3606	9128	3789
720 x 576	19488	7344	19515	7502	19791	7721
800 x 640	21648	8155	21701	8276	21967	8506
1280 x 720	36134	13426	36293	13548	36522	13765
1920 x 1080	55732	20707	55761	20823	56932	21059
2048 x 1536	61495	22816	61525	22890	62853	23164

7.5 A Memory-Efficient Parallel Architecture

In embedded vision systems, parallel computation of the integral image presents several design challenges in terms of hardware resources, speed and power consumption. Although recursive equations significantly reduce the number of operations for computing the integral image, the required internal memory becomes prohibitively large for an embedded integral image computation engine for increasing image sizes. With the objective of achieving high throughput with low hardware resources, this section proposes a memory-efficient design strategy for a parallel embedded integral image computation engine. Results show that the design achieves nearly 35% reduction in memory usage for common HD video.

Both the recursion-based serial [132] and parallel methods (in Section 7.3 and Section 7.4) require one complete row of integral image values to be stored in an internal memory so that it can be utilized for the calculation of the very next row. The width of the required internal memory is log_2(*number of rows* x *number of columns* x *maximum image pixel value*) rounded to the upper integer whereas the depth is equal to the total number of columns in one row of the image. Figure 7-7 highlights the internal memory requirements for an integral image computation engine implemented in hardware for some common images sizes. It is evident that with the increasing image size, the design of the integral image computation engine becomes inefficient in terms of hardware resources due to the large internal memory. It is desirable to achieve a design which is memory-efficient and provides high throughput.

To address the internal memory problem discussed above, a resource-efficient architecture is presented that is also capable of achieving high throughput. The design strategy makes use of the fact that integral image values in adjacent columns of a single row differ by a column sum (Figure 7-8). This difference value is maximum in the last row as the column sum includes all pixel values from the top to the bottom of the image in a particular column. In the worst case, the difference between two adjacent

columns in the last row of the image will be the product of the number of rows and the maximum value that can be attained by an image pixel (e.g., the maximum value is 255 for an 8-bit pixel).

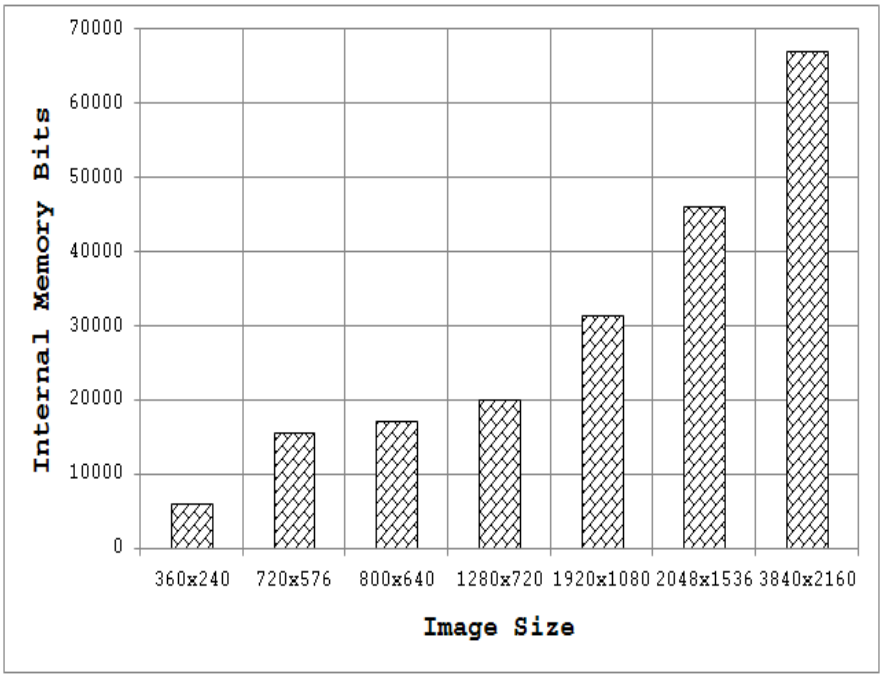

Figure 7-7: Internal memory requirements for the integral image computation engine for some common image sizes

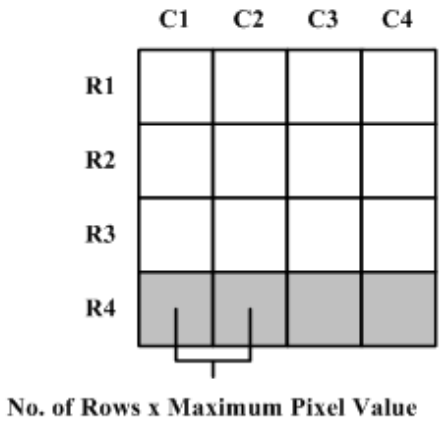

Figure 7-8: Worst case difference between adjacent integral image values in one row

Figure 7-9 shows a proposed architecture for an embedded integral image computation engine. This pipelined architecture computes two integral image values in a single clock cycle. Unlike the parallel methods

presented in Section 7.3 and Section 7.4 which store a complete row of integral image values in internal memory for computing the next row, this design strategy saves only the difference values of the adjacent columns in a row for calculating the next row. Only the integral image value for the first column in that row is saved in a separate register to allow computation of the integral image values from the stored difference values. Although the depth of the internal memory remains the same as mentioned above, the proposed design approach requires the width to be log_2(*number of rows* x *maximum image pixel value*) rounded to the upper integer value. Table 7-2 provides the results for internal memory reduction when prototyped on an FPGA, a Virtex-6 XC6VLX240T device, for some common image sizes. The maximum frequency of the design is 146.71 MHz. It is evident from Table 7-2 that the architecture is capable of achieving significant memory reduction over other recursion-based methods, even for small image sizes.

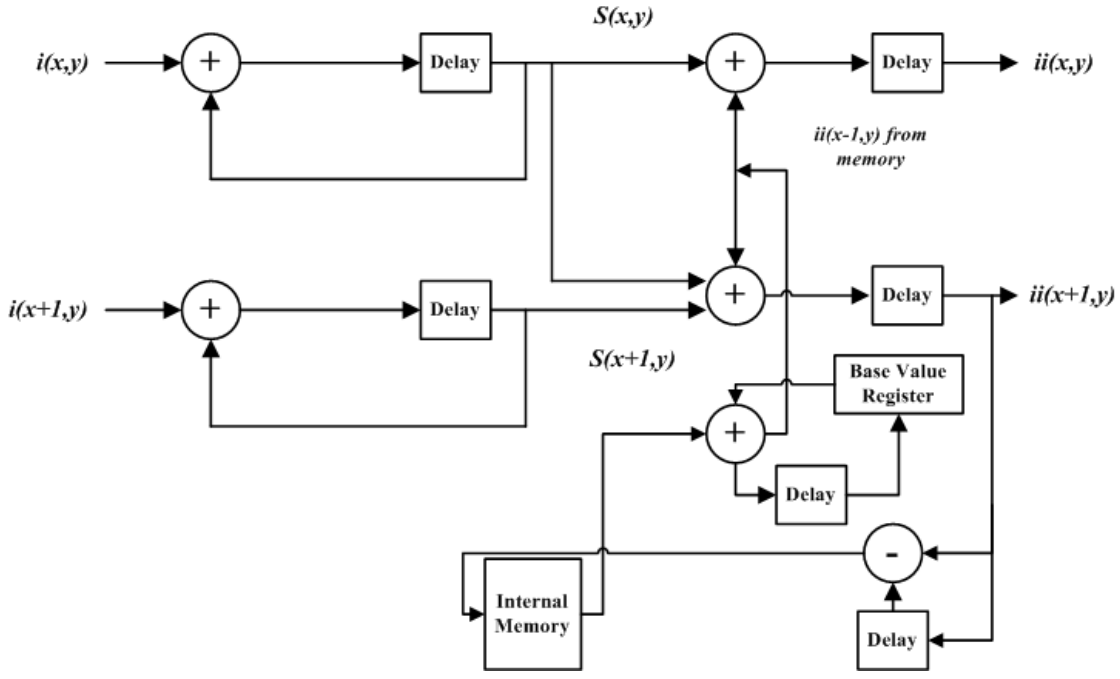

Figure 7-9: Block diagram of the proposed architecture. i(x,y) and ii(x,y) are the image pixel value and the integral image value at location (x,y) in the image. S(x,y) is the row sum at that particular location

Table 7-2: Reduction in internal memory requirements for the Proposed Architecture on Virtex-6 XC6VLX240T

Image Size	Memory-Efficient Design Strategy		Reduction in Internal Memory Bits relative to other recursion based methods	Reduction in resource consumption relative to 2-Rows Algorithm	
	Slice Registers	LUTs		Slice Registers	LUTs
360 x 240	6307	2792	32%	30.50%	22.57%
720 x 576	13164	5537	33.3%	32.54%	26.19%
800 x 640	14602	6047	33.3%	32.71%	26.93%
1280 x 720	24668	9864	32.1%	32.03%	27.19%
1920 x 1080	37145	14614	34.4%	33.39%	29.82%
2048 x 1536	39694	15558	36.6%	35.48%	32.03%

7.6 Efficient Storage of Integral Image

As opposed to its computation, storage of the integral image has received less attention until recently. The only work of significance in this domain is presented in [243]. Memory requirements for an integral image are substantially larger than for the input image. For resource-constrained embedded vision systems, storage of the integral image presents several design challenges. In this section, two viable techniques for reducing the memory requirements of an integral image are proposed for different application scenarios. Both hardware and software solutions can benefit from the presented techniques. Results for some common image sizes are presented which show that the methods guarantee a minimum of 44.44% reduction in memory for all image sizes and application scenarios, and may achieve reduction of more than 50% in specific situations for embedded vision systems.

The bars in Figure 7-10 show the storage requirements of an integral image for some common image sizes (read values from the left ordinate axis), while the line indicates the percentage increase in memory with respect to the input image considering 8-bit pixels (read values from the right ordinate axis). It is evident from Figure 7-10 that the storage requirements of the integral image are much larger than for the input image. Since there is a corresponding integral image value for every pixel in

the input image, the dimensions of the two image representations are the same. The increased memory requirement for the integral image is therefore a consequence of the much larger word length of the integral image values as compared to the input pixel values (which are usually 8-bit wide). Figure 7-11 depicts the word lengths required for an integral image considering 8-bit input pixel values for some common image sizes. It can be seen clearly that with increasing image size, the required word length for the integral image also increases.

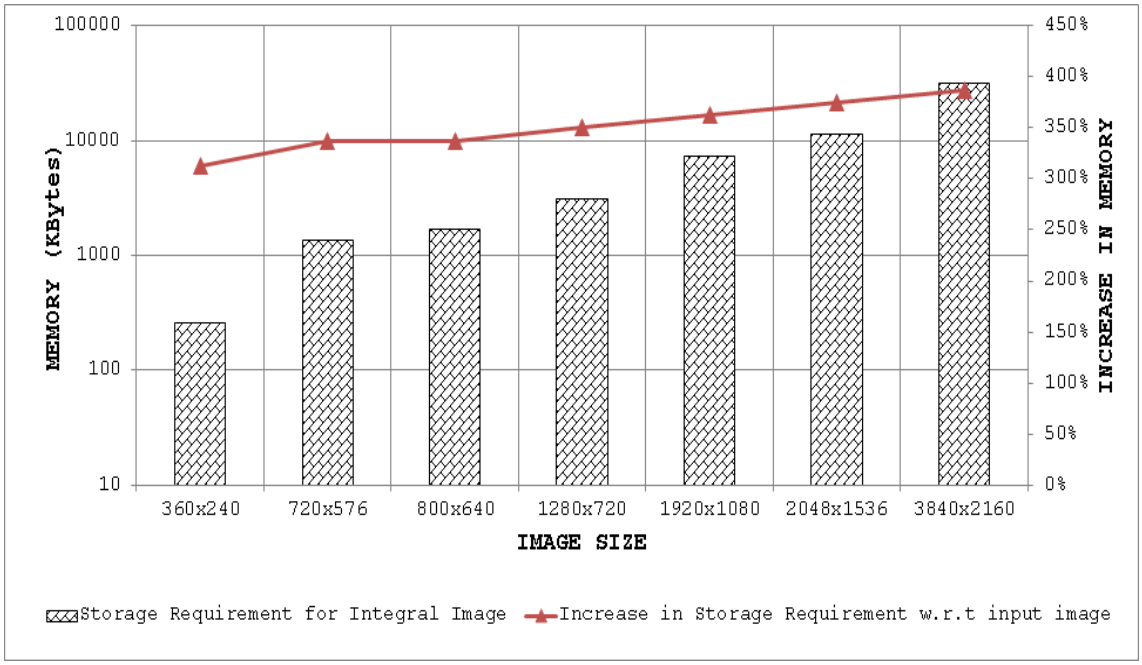

Figure 7-10: Storage requirements of the integral image for some common image sizes and percentage increase in memory relative to the input image (considering 8-bit pixels)

7.6.1 Limitations of Existing Methods

Although the exact and approximate methods presented in [243] manage to reduce the word length of an integral image, they do have some limitations:

1) These methods are applicable only in situations where the size of the box filter is *a priori* known.

2) The exact method achieves negligible reduction in memory if the maximum size of the box filter is almost equal to the input image size.

3) The approximate method involves loss of accuracy due to rounding pixel values. For example, there is significant increase in false detection rate for the Viola-Jones face detector when the approximate method is used in [243].

4) Although the exact method does not incur any loss of accuracy, it fails to achieve significant reduction in word length.

5) These techniques are one-dimensional in the sense that they only concentrate only on reducing the word length of the integral image, which in turn affects the width of the storage memory but not its depth.

To overcome the above-mentioned shortcomings, two methods are presented for storing the integral image efficiently in embedded vision systems without any loss of accuracy. The first of these is appropriate for any application that involves an integral image without prior knowledge of the box filter size and in situations where the size of the box filter is nearly the same as that of the input image. The second method is suited to applications where the size of box filter is *a priori* known (*e.g.,* SURF [13]).

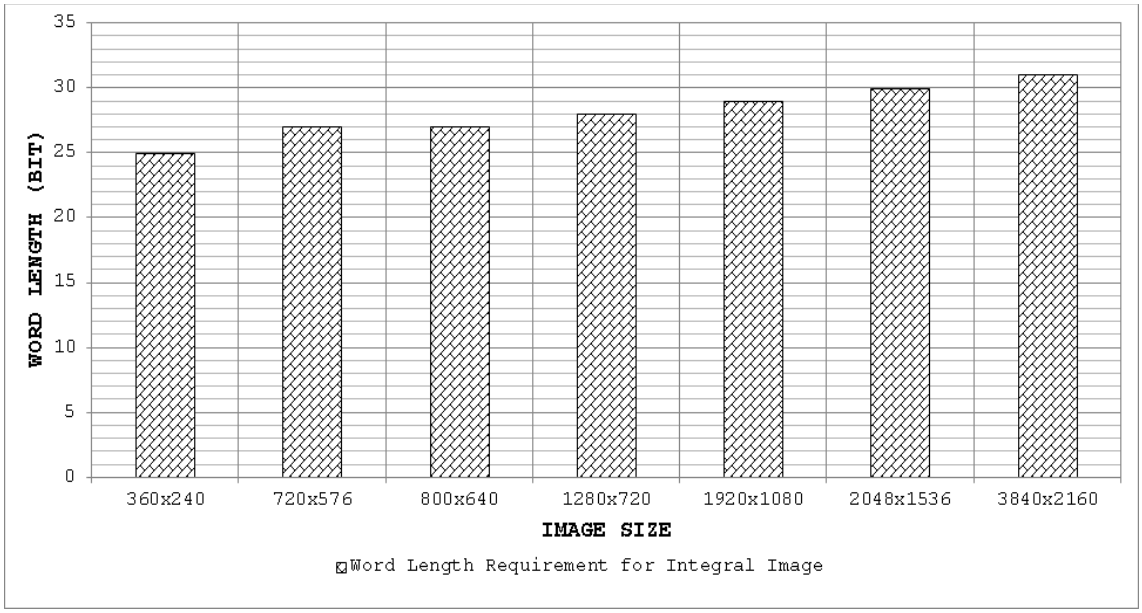

Figure 7-11: Word length requirements for integral image for some common image sizes considering 8-bit input pixels

7.6.2 Proposed Method 1

This is a general technique that guarantees 44.44% reduction in memory resources for storing an integral image and can be utilized for any application involving integral images. The method is lossless and is suitable for scenarios where the box filter size is either unknown or is not much smaller than the image size. The technique is especially attractive for embedded systems, as the same system can be utilized for different applications without any modifications to hardware or software.

Unlike the methods in [243], the proposed technique attempts to reduce the depth of the memory required to store an integral image. For this particular method, the width of the memory is assumed to be log_2(*length of the image* x *width of the image* x *maximum pixel value*) rounded to the upper integer value. The first step is to make the length and width of the integral image both into multiples of 3. For example, if the integral image dimensions are 360 x 240 then the length and the width values are already multiples of 3 and nothing needs to be done. Otherwise, the last rows and/or columns of the integral image are discarded to achieve this objective. In the worst case, the last two rows and the last two columns need to be eliminated. The whole integral image is then divided into blocks of 3 x 3 integral image values. Figure 7-12 depicts a single such block. The shaded integral images values in Figure 7-12 are the ones that are selected by the proposed method to store in the memory; the remaining four values on the corners are discarded. Despite not storing these four corner integral image values, the 3 x 3 integral image block can be perfectly reconstructed from the stored integral image values by utilizing the fact that

$$a = b + d - e + input\ pixel\ value\ at\ e \qquad \textit{Equation 7-20}$$

$$c = b + f - e - input\ pixel\ value\ at\ f \qquad \textit{Equation 7-21}$$

$$g = d + h - e - input\ pixel\ value\ at\ h \qquad \textit{Equation 7-22}$$

$$i = h + f - e + input\ pixel\ value\ at\ i \qquad \textit{Equation 7-23}$$

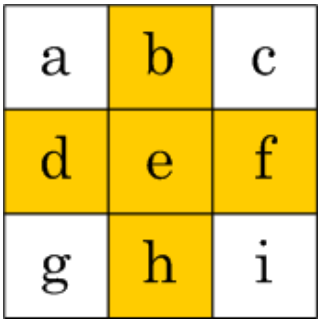

Figure 7-12: A sample 3x3 integral image block for the proposed method. The shaded region shows the integral image values that need to be stored

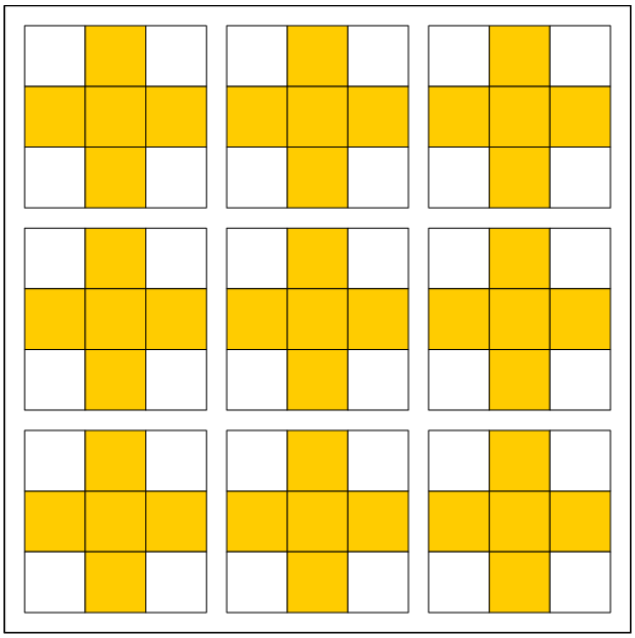

Figure 7-13: A sample integral image of dimensions 9 x 9. The shaded regions indicate the integral image values that need to be stored in the memory

Figure 7-13 shows all 3 x 3 blocks for a sample integral image of dimensions 9 x 9 (with values shaded as to whether they need to be stored or discarded). Out of the 81 integral image values in Figure 7-13, only 45 values need to be stored in memory, meaning that this method achieves a 44.44% reduction in the storage requirements for the integral image. Moreover, this reduction is independent of the input image size and the box filter size.

As a box type filter can be computed quickly using three addition and subtraction operations when the integral image values on the four corners of that filter are known [132] (see Figure 7-14), the proposed method does not require any extra computation if the required four values are those which are stored in memory. In the worst case, all four integral image values needed for computing the box filter will not be available from memory. In that particular case, Equation 7-20 to Equation 7-23 can be utilized for computing the integral image values which were discarded earlier; they can then be used for calculating the required box filter. Although there is a speed-memory tradeoff involved, the method is still an efficient way of computing box type filters as it eliminates computation intensive multiplications.

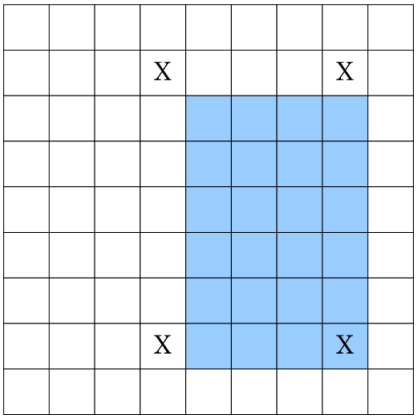

Figure 7-14: Box filter calculation using the integral image; the shaded area indicates the filter to be computed whereas 'X' shows the integral image values required for computation of this box filter

7.6.3 Proposed Method 2

In an effort to reduce the size of the memory required for storing the integral image further, a technique is presented here which decreases *both* the width and the depth of memory. It combines the exact method presented in [243] with the technique proposed in Section 7.6.2. This hybrid method is suitable for scenarios where the maximum size of the box filter to be computed is considerably smaller than the input image size. Again, the method does not incur any loss of accuracy.

The worst-case integral image value that determines the binary word length required to represent integral image is dependent upon the width, height and number of bits per pixel of the input image. This can be stated as [243]:

$$ii_{max} = (2^{L_i} - 1) \text{ x } W \text{ x } H \qquad \textit{Equation 7-24}$$

where i is the input image, ii is the integral image, ii_{max} is the worst case integral image value, W is the width of the input image, H is the height of the input image and L_i is the number of bits per pixel for the input image. According to [243], the number of bits L_{ii} required for representing the worst case integral image value thus needs to satisfy

$$(2^{L_{ii}} - 1) \geq (2^{L_i} - 1) \text{ x } W \text{ x } H \qquad \textit{Equation 7-25}$$

The total memory in bytes required to store the integral image can be calculated as follows:

$$Memory = \frac{(W \text{ x } H) \text{ x } L_{ii}}{8} \qquad \textit{Equation 7-26}$$

According to the exact method in [243], for platforms with complement-coded arithmetic, if the maximum height and the width of the box filter to be calculated are known, then the word length for the integral image needs to satisfy:

$$(2^{L_{ii}} - 1) \geq (2^{L_i} - 1) \text{ x } W_{max} \text{ x } H_{max} \qquad \textit{Equation 7-27}$$

where W_{max} is the maximum width of box filter and H_{max} is the maximum height of a box filter. Equation 7-27 can be explained on the basis that if a chain of linear operations is performed on integers and there are some intermediate overflowing results then it is possible to get the correct final result if this result can be represented by the used data word length [243]. The proposed method first makes both the length and width of the input image multiples of 3. Equation 7-27 is then used to find the required word length for storing the integral image. As a final step, the depth of the memory is reduced by employing the method proposed in Section 7.6.2.

A variant of this method can also prove useful. Observing Equation 7-27 closely reveals that the supposition of having all pixel values in the input image set to their maximum value (255 in the case of 8-bit pixels) for evaluating the worst case integral image value does not seem very practical for feature detection techniques like SURF which is used for blob detection. i.e., to detect dark areas/regions in the input image surrounded by light ones or vice versa. Assuming that all the pixels are set to their maximum value in the input image implies that there is absolutely no variation in the pixel values. Since most feature detection techniques try to detect features in those areas of the image where there are large changes in pixel values, this assumption simply means that there are no features to be detected in the input image.

This variant of the above technique further extends the exact method of [243] by supposing that there is variation in pixel values of the input image. Equation 7-27 is thus modified as:

$$(2^{L_{ii}} - 1) \geq [(2^{L_i} - 1) \text{ x } (W_{max} \text{ x } H_{max}) \text{ x } 0.96 + (2^{L_i - 1} - 1) \text{ x } (W_{max} \text{ x } H_{max}) \text{ x } 0.04] \quad \textit{Equation 7-28}$$

It is assumed here that 96% of all pixels in a box filter to be evaluated have maximum values, while the other 4% of pixels have half the maximum value. This is a suitable approximation as most images generally have more variation in pixel values than given by Equation 7-28. The final step is to reduce the depth of the required memory by employing the technique presented in Section 7.6.2.

Figure 7-15 shows comparative results for the two variants of the proposed method and the original exact method [243] for the specific case of the SURF detector with increasing image sizes by taking $W_{max} = 65$ and $H_{max} = 129$. Note that the largest box filter to be computed for SURF is 195 x 195 but it can be broken down into three box type filters of 65 x 129 (or 129 x 65)[13]. The bars in Figure 7-15 represent the memory required for storing the integral image (read values from the left ordinate axis) whereas

the line graphs show the percentage reduction in memory (read values from the right ordinate axis) relative to the actual requirement (see Figure 7-10). It is evident that the best performance in terms of memory reduction is achieved by utilizing Equation 7-28 in combination with the depth reduction method from Section 7.6.2 (Variant 2 in Figure 7-15). It can be seen that the two variants of the proposed method out-perform the original exact method comprehensively and allow more than 50% reduction in memory, even for small sized images.

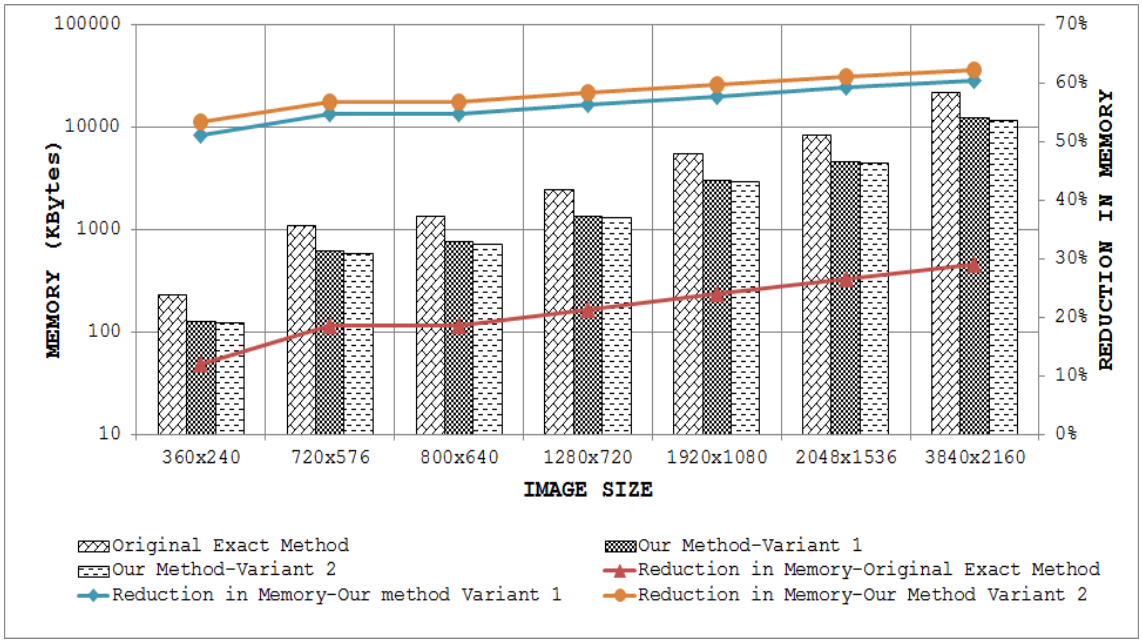

Figure 7-15: Comparative results for the original exact method [243] and the two variants of the proposed technique

7.7 Summary

This chapter has addressed computation and storage issues related to integral images. It has analyzed integral image calculation from a parallel computation perspective. With the objective of reducing computational resources, two hardware algorithms based on the decomposition of the Viola-Jones recursive equations were proposed in this chapter. These are capable of providing up to four integral image values per clock cycle without any significant increase in the number of addition operations. An efficient design strategy for a parallel embedded integral image computation engine

that is capable of achieving nearly 35% reduction in internal memory for common HD video (1920 x 1080) was also proposed. Finally, the chapter has presented two methods for the reduction of memory for storing an integral image. These techniques guarantee at least 44.44% reduction in memory and may allow more than 50% reduction when the maximum size of a box filter to be computed is considerably smaller than the input image size.

8 Conclusions and Future Directions

I start where the last man left off.

THOMAS EDISON

A summary of the main contributions made by this thesis is provided in this chapter. Some potential future directions are also suggested for building on its work. These are followed by closing remarks to end the chapter.

8.1 Summary of Contributions

Determining image correspondences is a fundamental problem in computer vision. This thesis generally targeted this problem while mainly concentrating on the detection step for improving the effectiveness and reliability of the three-stage system (consisting of detection, description and matching shown in Figure 1-1). A brief account of the major contributions of this thesis is given below.

- Predicting the performance of vision-based systems which usually operate in complex and unknown environments across a range of different applications is a challenging task. The metrics which are currently available to gauge the performance of feature detectors do not always reflect actual performance. Moreover, existing performance measures allow only offline assessment due to their requirement for ground truth data and high computation cost. To bridge these research gaps, Chapters 3, 4 and 5 presented offline and online performance metrics that reflect real-world performance for local feature detectors. The proposed online metrics can be computed quickly and allow a feature detector to gauge its own performance and take appropriate actions online to maximize its effectiveness by adapting to the nature of the imagery it is processing. Furthermore, the thesis has shown how these metrics can be utilized for building more effective vision systems, confirming in a statistically meaningful way that these metrics work.

- In vision research, absolute and relative evaluation of feature detectors is considered an important issue as it is useful for selecting a particular detector for some specific application depending on its strengths and weaknesses. This thesis has presented comparative results for several state-of-the-art detectors on standard datasets and some newly acquired large image databases utilizing the designed performance metrics. The presented results contradict some previous

findings and have advanced current understanding about the behavior of these feature detectors.

- Running multiple feature detectors simultaneously to tackle the uncertainty of image content for solving complex vision problems has detrimental effects on overall computation time, usually provides an overcomplete representation of an image rather than a compact one, and may have adverse effects on the combined performance of multiple detectors. To address this issue, this thesis has shown how knowledge of individual feature detectors' functions allows them to be combined efficiently and made into an integral part of a robust vision system (see Chapter 5).

- Many state-of-the-art feature detection techniques are unsuitable for real-time applications on commodity hardware and show poor matching performance in the presence of various image transformations. To tackle these problems, this thesis has presented several improvements to feature detectors' performance in terms of matching accuracy and speed of execution (see Chapters 4, 6 and 7).

- Embedded systems generally have strict constraints on computational resources, power consumption, memory size, chip area and weight; this makes the task of running computation-intensive feature detection algorithms on such systems challenging. To that end, Chapter 7 demonstrated how resource-efficient architectures can be designed for local feature detection methods by focusing on the efficient computation and storage of the integral image.

8.2 Future Directions

The thesis has shown how the improved repeatability metrics (presented in Chapter 3) can be employed for the framework (proposed in Chapter 4) for vision system design under blur, JPEG compression and uniform light variations. This work can be extended to develop new datasets for scale,

rotation and viewpoint changes with a large number of different scene types and utilize the proposed framework for establishing the upper and lower performance bounds of state-of-the-art detectors and to identify statistically-significant performance differences between them as a function of the amount of image transformation.

Another refinement of the work presented in this thesis would be the evaluation of the combined effect of different image transformations on the performance of state-of-the-art feature detectors utilizing the framework presented in Chapter 4. For example, it will be valuable to see how a feature detection technique performs when blur and viewpoint changes occur simultaneously. This type of approach will make analysis more thorough and much closer to the real-world situations where one often encounters multiple types of transformation in a pair of images. This will be really useful from a vision systems design perspective.

Without any doubt, the human visual system (HVS) is the ultimate vision system. To compare the performances of state-of-the-art feature detectors utilizing the improved repeatability metrics proposed in Chapter 3 while considering the HVS as reference is another promising direction. Such work will identify detectors that find feature points mostly in those areas of the input image which are considered interesting by the human eye.

In Chapter 5, it was demonstrated that utilizing the coverage metrics, one can combine feature detectors intelligently. A possible extension can be the incorporation of the coverage metric in the design of an individual detector for selecting only those points that increase the overall coverage. Moreover, such individual detectors can also be combined to further improve performance. The proposed prediction-based framework can also be extended by making it adaptive. This essentially means that the system would initially utilize the general guidelines for combining feature detectors stored in a database (as given in Chapter 5) but will also keep track of the

actual performance it is achieving following them and would be able to update the database based on these observations.

Another potential future direction is to design a low-power, high-throughput architecture for SURF detector utilizing the integral image computation and storage algorithms of Chapter 7.

Since SFOP detector shows good performance in the quantitative evaluations done in Chapters 4 and 5, a promising future direction would be to improve its performance further and investigate efficient computation architectures for this algorithm.

8.3 Closing Remarks

We need better features.

[5]DAVID G. LOWE, 2009

Ten years after proposing the highly influential SIFT algorithm, this perspicacious statement from David Lowe clearly reflects that there is still plenty of room for improvement in the domain of local invariant feature detection. Although this thesis has endeavored to push the boundaries further, many challenges lie ahead for the vision community in the context of overall advancement of the field. Surely, the words of David Lowe will continue to befit this situation until the emergence of *next SIFT*.

– THE END –

[5] 'Some Machine Learning Problems that We in the Computer Vision Community would like to see solved', Keynote speech by William T. Freeman, MIT at NIPS 2009 Workshop on Approximate Learning of Large Scale Graphical Models: Theory and Applications, Whistler, Canada, December 12, 2009.

Bibliography

[1] T. Tuytelaars, and K. Mikolajczyk, “Local Invariant Feature Detectors: A Survey,” *Foundations and Trends in Computer Graphics and Vision,* vol. 3, no. 3, pp. 177-280, 2007.

[2] F. Attneave, “Some Informational Aspects of Visual Perception,” *Psychological Review,* vol. 61, pp. 183-193, 1954.

[3] H. Moravec, "Towards Automatic Visual Obstacle Avoidance," *Proc. International Joint Conference on Artificial Intelligence*, pp. 584-590, 1977.

[4] C. Harris, and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Alvey Vision Conference*, pp. 147-151, 1988.

[5] P. R. Beaudet, "Rotationally Invariant Image Operators," *Proc. International Joint Conference on Pattern Recognition*, pp. 579-583, 1978.

[6] L. Kitchen, and A. Rosenfeld, “Gray-Level Corner Detection,” *Pattern Recognition Letters,* vol. 1, pp. 95-102, 1982.

[7] L. Dreschler, and H. H. Nagel, “Volumetric Model and 3D Trajectory of a Moving Car Derived from Monocular TV Frame Sequences of a Street Scene,” *Computer Graphics and Image Processing,* vol. 20, pp. 199-228, 1982.

[8] W. Forstner, and E. Gulch, "A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features," *Proc. ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data*, pp. 281-305, 1987.

[9] W. Forstner, "A Framework for Low Level Feature Extraction," *Proc. The 3rd European Conference on Computer Vision*, pp. 383-394, Stockholm, Sweden, 1994.

[10] S. Smith, and J. Brady, “SUSAN—A New Approach to Low Level Image Processing,” *International Journal of Computer Vision,* vol. 23, no. 1, pp. 45-78, 1997.

[11] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. International Conference on Computer Vision (ICCV)*, pp. 1150-1157, Corfu, Greece, 1999.

[12] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding,* vol. 110, no. 3, pp. 346-359, 2008.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Proc. The British Machine Vision Conference*, pp. 384-393, Cardiff, UK, 2002.

[15] K. Mikolajczyk, and C. Schmid, “Scale & Affine Invariant Interest Point Detectors,” *International Journal of Computer Vision,* vol. 60, no. 1, pp. 63-86, 2004.

[16] W. Forstner, T. Dickscheid, and F. Schindler, "Detecting Interpretable and Accurate Scale-Invariant Keypoints," *Proc. The 12th IEEE International Conference on Computer Vision*, pp. 2256-2263, Kyoto, Japan, 2009.

[17] E. Rosten, R. Porter, and T. Drummond, “Faster and Better: A Machine Learning Approach to Corner Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 1, pp. 105-119, 2010.

[18] T. Kadir, A. Zisserman, and M. Brady, "An Affine Invariant Salient Region Detector," *Proc. The 8th European Conference on Computer Vision*, pp. 228-241, Prague, 2004.

[19] T. Dickscheid, F. Schindler, and W. Förstner, “Coding Images with Local Features,” *International Journal of Computer Vision,* vol. 94, no. 2, pp. 154-174, 2010.

[20] W. Kienzle, F. A. Wichmann, B. Scholkopf, and M. O. Franz, "Learning an Interest Operator from Human Eye Movements," *Proc. Conference on Computer Vision and Pattern Recognition Workshop*, USA, 2006.

[21] V. Lepetit, and P. Fua, “Keypoint Recognition using Randomized Trees,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 9, pp. 1465-1479, 2006.

[22] E. Rosten, and T. Drummond, "Fusing Points and Lines for High Performance Tracking," *Proc. The 10th IEEE International Conference on Computer Vision*, pp. 1508-1515, Beijing, China, 2005.

[23] E. Rosten, and T. Drummond, "Machine Learning for High-Speed Corner Detection," *Proc. European Conference on Computer Vision*, pp. 430-443, 2006.

[24] L. Trujillo, and G. Olague, “Automated Design of Image Operators that Detect Interest Points,” *Evolutionary Computation,* vol. 16, no. 4, pp. 483-507, 2008.

[25] L. Trujillo, and G. Olague, "Synthesis of Interest Point Detectors through Genetic Programming," *Proc. The 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 887-894, Seattle, WA, USA, 2006.

[26] M. Brown, and D. G. Lowe, "Recognising Panoramas," *Proc. International Conference on Computer Vision (ICCV)*, pp. 1218-1227, 2003.

[27] M. Brown, and D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *International Journal of Computer Vision,* vol. 74, no. 1, pp. 59-77, 2007.

[28] T. Tuytelaars, and L. Van Gool, "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions," *Proc. The 11th British Machine Vision Conference*, pp. 412-425, Bristol, UK, 2000.

[29] T. Tuytelaars, and L. Van Gool, "Content-Based Image Retrieval Based on Local Affinely Invariant Regions," *Proc. International Conference on Visual Information Systems*, pp. 493-500, 1999.

[30] J. Sivic, and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. ICCV*, pp. 1470-1477, Nice, France, 2003.

[31] J. Knopp, J. Sivic, and T. Pajdla, "Avoiding Confusing Features in Place Recognition," *Proc. European Conference on Computer Vision (ECCV)*, pp. 748-761, 2010.

[32] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints," *Proc. ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[33] G. Dorkó, and C. Schmid, "Selection of Scale-Invariant Parts for Object Class Recognition," *Proc. The 9th IEEE International Conference on Computer Vision*, pp. 634-639, Nice, France, 2003.

[34] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 264-271, Madison, Wisconsin, USA, 2003.

[35] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "Weak Hypotheses and Boosting for Generic Object Detection and Recognition," *European Conference on Computer Vision*, pp. 71-84, 2004.

[36] A. Turina, T. Tuytelaars, and L. Van Gool, "Efficient Grouping Under Perspective Skew," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 247-254, Hawaii, USA, 2001.

[37] S. Se, D. Lowe, and J. Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *The International Journal of Robotics Research,* vol. 21, no. 8, pp. 735-758, 2002.

[38] F. Schaffalitzky, and A. Zisserman, "Automated Location Matching in Movies," *Computer Vision and Image Understanding,* vol. 92, no. 2, pp. 236-264, 2003.

[39] S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation using Affine-Invariant Regions," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 319-324, Madison, Wisconsin, USA, 2003.

[40] S. Lazebnik, C. Schmid, and J. Ponce, "Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition," *Proc. The 9th IEEE International Conference on Computer Vision*, pp. 649-655, Nice, France, 2003.

[41] C. C. Wang, and K. C. Wang, "Hand Posture Recognition using Adaboost with SIFT for Human Robot Interaction," *Recent Progress in Robotics: Viable Robotic Service to Human*, pp. 317-329, 2008.

[42] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the World from Internet Photo Collections," *International Journal of Computer Vision,* vol. 80, no. 2, pp. 189-210, 2008.

[43] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Learning Hierarchical Models of Scenes, Objects, and Parts," *Proc. IEEE International Conference on Computer Vision*, Beijing, China, 2005.

[44] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, *Discovering Object Categories in Image Collections, Technical Report*, Massachusetts Institute of Technology, 2005.

[45] T. Dickscheid, and W. Förstner, "Evaluating the Suitability of Feature Detectors for Automatic Image Orientation Systems," *Proc. ICVS*, pp. 305-314, Liege, Belgium, 2009.

[46] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, "A Comparison of Affine Region Detectors," *International Journal of Computer Vision,* vol. 65, no. 1, pp. 43-72, 2005.

[47] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of Interest Point Detectors," *International Journal of Computer Vision,* vol. 37, no. 2, pp. 151-172, 2000.

[48] P. J. Phillips, and K. W. Bowyer, "Introduction to the Special Section on Empirical Evaluation of Computer Vision Algorithms," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 289-290, 1999.

[49] A. Torralba, and A. A. Efros, "Unbiased Look at Dataset Bias," *Proc. CVPR*, pp. 1521-1528, USA, 2011.

[50] K. Mikolajczyk. "Oxford Data Sets," http://www.robots.ox.ac.uk/~vgg/research/affine/.

[51] Q. McNemar, "Note on the Sampling Error of the Difference between Correlated Proportions or Percentages," *Psychometrika,* vol. 12, no. 2, pp. 153-157, 1947.

[52] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical Methods for Rates and Proportions*, Third ed.: John Wiley & Sons, 2003.

[53] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," *Proc. 9th European Conference on Computer Vision*, pp. 404-417, Graz, Austria, 2006.

[54] Y. Ke, and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 506-513, USA, 2004.

[55] K. Mikolajczyk, and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, 2005.

[56] N. Ansari, and E. J. Delp, "On Detecting Dominant Points," *Pattern Recognition,* vol. 24, no. 5, pp. 441-451, 1991.

[57] H. Lynn Beus, and S. S. H. Tiu, "An Improved Corner Detection Algorithm Based on Chain-Coded Plane Curves," *Pattern Recognition,* vol. 20, no. 3, pp. 291-296, 1987.

[58] S. C. Pei, and J. H. Horng, "Corner Point Detection using Nest Moving Average," *Pattern Recognition,* vol. 27, no. 11, pp. 1533-1537, 1994.

[59] J. Cooper, S. Venkatesh, and L. Kitchen, "Early Jump-Out Corner Detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, no. 8, pp. 823-828, 1993.

[60] D. J. Beymer, "Finding Junctions using the Image Gradient," *Proc. International Conference on Computer Vision and Pattern Recognition*, pp. 720-721, 1991.

[61] I. M. Anderson, and J. C. Bezdek, "Curvature and Tangential Deflection of Discrete Arcs: A Theory based on the Commutator of Scatter Matrix Pairs and its Application to Vertex Detection in

Planar Shape Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 6, pp. 27-40, 1984.

[62] H. Asada, and M. Brady, "The Curvature Primal Sketch," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 1, pp. 2-14, 1986.

[63] J. G. Dunham, "Optimum Uniform Piecewise Linear Approximation of Planar Curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 1, pp. 67-75, 1986.

[64] H. Freeman, "Computer Processing of Line-Drawing Images," *ACM Computing Surveys,* vol. 6, no. 1, pp. 57-97, 1974.

[65] H. Freeman, and L. S. Davis, "A Corner-Finding Algorithm for Chain-Coded Curves," *IEEE Transactions on Computers* vol. 26, pp. 297-303, 1977.

[66] X. C. He, and N. H. C. Yung, "Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of support," *Proc. International Conference on Pattern Recognition*, pp. 791-794, 2004.

[67] R. Horaud, F. Veillon, and T. Skordas, "Finding Geometric and Relational Structures in an Image," *European Conference on Computer Vision*, pp. 374-384, 1990.

[68] Q. Ji, and R. M. Haralick, "Corner Detection with Covariance Propagation," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 362-367, 1997.

[69] D. Langridge, "Curve Encoding and the Detection of Discontinuities," *Computer Graphics and Image Processing,* vol. 20, pp. 58-71, 1982.

[70] T. Lindeberg, "Edge Detection and Ridge Detection with Automatic Scale Selection," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 465-470, 1996.

[71] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision,* vol. 30, no. 2, pp. 79-116, 1998.

[72] D. G. Lowe, "Organization of Smooth Image Curves at Multiple Scales," *International Journal of Computer Vision,* vol. 3, no. 2, pp. 119-130, 1989.

[73] G. Medioni, and Y. Yasumoto, "Corner Detection and Curve Representation using Cubic B-splines," *Computer Vision, Graphics, and Image Processing,* vol. 39, no. 3, pp. 267-278, 1987.

[74] F. Mokhtarian, and A. Mackworth, "Scale-based Description and Recognition of Planar Curves and Two-dimensional shapes," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 1, pp. 34-43, 1986.

[75] F. Mokhtarian, and R. Suomela, "Robust Image Corner Detection through Curvature Scale Space," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 12, pp. 1376-1381, 1998.

[76] H. Ogawa, "Corner Detection on Digital Curves based on Local Symmetry of the Shape," *Pattern Recognition,* vol. 22, no. 4, pp. 351-357, 1989.

[77] A. Rattarangsi, and R. T. Chin, "Scale-based Detection of Corners of Planar Curves," *Proc. International Conference on Pattern Recognition*, pp. 923-930 1990.

[78] A. Rosenfeld, and E. Johnston, "Angle Detection on Digital Curves," *IEEE Transactions on Computers,* vol. C-22, no. 9, pp. 875-878, 1973.

[79] A. Rosenfeld, and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," *IEEE Transactions on Computers,* vol. C-20, no. 5, pp. 562-569, 1971.

[80] A. Rosenfeld, M. Thurston, and Y. H. Lee, "Edge and Curve Detection: Further Experiments," *IEEE Transactions on Computers,* vol. C-21, no. 7, pp. 677-715, 1972.

[81] A. Rosenfeld, and J. S. Weszka, "An Improved Method of Angle Detection on Digital Curves," *IEEE Transactions on Computers,* vol. C-24, no. 9, pp. 940-941, 1975.

[82] P. L. Rosin, "Representing Curves at their Natural Scales," *Pattern Recognition,* vol. 25, no. 11, pp. 1315-1325, 1992.

[83] P. L. Rosin, "Determining Local Natural Scales of Curves," *Pattern Recognition Letters,* vol. 19, no. 1, pp. 63-75, 1998.

[84] W. S. Rutkowski, and A. Rosenfeld, *A Comparison of Corner Detection Techniques for Chain Coded Curves, Technical Report*, University of Maryland, 1978.

[85] J. Fang, and T. S. Huang, "A Corner Finding Algorithm For Image Analysis and Registration," *Proc. AAAI Conference*, pp. 46-49, 1982.

[86] 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,* vol. 15, no. 6, pp. 415-434, 1997.

[87] H. P. Moravec, "Visual Mapping by a Robot rover," *Proc. International Joint Conference on Artificial Intelligence*, pp. 598-600, 1979.

[88] H. P. Moravec, "Rover Visual Obstacle Avoidance," *Proc. International Joint Conference on Artificial Intelligence*, pp. 785-790, 1981.

[89] A. Baumberg, "Reliable Feature Matching across Widely Separated Views," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 774-781, 2000.

[90] M. H. Chen, and P. F. Yan, "A Multiscanning Approach based on Morphological Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 694-700, 1989.

[91] C. H. Chen, J. S. Lee, and Y. N. Sun, "Wavelet transformation for Gray-Level Corner Detection," *Pattern Recognition,* vol. 28, no. 6, pp. 853-861, 1995.

[92] Y. Dufournaud, C. Schmid, and R. Horaud, "Matching Images with Different Resolutions," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 612-618, 2000.

[93] C. S. Kenney, B. Manjunath, M. Zuliani, G. A. Hewer, and A. Van Nevel, "A Condition Number for Point Matching with Application to Registration and PostRegistration Error Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, no. 11, pp. 1437-1454, 2003.

[94] C. Kenney, M. Zuliani, and B. Manjunath, "An Axiomatic Approach to Corner Detection," *Proc. International Conference on Computer Vision and Pattern Recognition*, pp. 191-197, 2005.

[95] R. Laganiere, "A Morphological Operator for Corner Detection," *Pattern Recognition,* vol. 31, no. 11, pp. 1643-1652, 1998.

[96] R. S. Lin, C. H. Chu, and Y. C. Hsueh, "A Modified Morphological Corner Detector," *Pattern Recognition Letters,* vol. 19, no. 3, pp. 279-286, 1998.

[97] J. A. Noble, "Finding Corners," *Image and Vision Computing,* vol. 6, no. 2, pp. 121-128, 1988.

[98] M. Trajković, and M. Hedley, "Fast Corner Detection," *Image and Vision Computing,* vol. 16, no. 2, pp. 75-87, 1998.

[99] F. Heitger, L. Rosenthaler, R. Von Der Heydt, E. Peterhans, and O. Kübler, "Simulation of Neural Contour Mechanisms: From Simple to End-Stopped Cells," *Vision Research,* vol. 32, no. 5, pp. 963-981, 1992.

[100] K. R. Cave, and J. M. Wolfe, "Modeling the Role of Parallel Processing in Visual Search," *Cognitive Psychology,* vol. 22, no. 2, pp. 225-271, 1990.

[101] P. Burt, and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications,* vol. 31, no. 4, pp. 532-540, 1983.

[102] R. Bajcsy, "Computer Identification of Visual Surfaces," *Computer Graphics and Image Processing,* vol. 2, no. 2, pp. 118-130, 1973.

[103] V. H. Brecher, R. Bonner, and C. Read, "Model of Human Preattentive Visual Detection of Edge Orientation Anomalies," *Proc. The SPIE Conference of Visual Information Processing: From Neurons to Chips*, pp. 39-51, 1991.

[104] S. Grossberg, E. Mingolla, and D. Todorovic, "A Neural Network Architecture for Preattentive Vision," *IEEE Transactions on Biomedical Engineering,* vol. 36, no. 1, pp. 65-84, 1989.

[105] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-based Visual Attention for Rapid Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1254-1259, 1998.

[106] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust Object Recognition with Cortex-like Mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 3, pp. 411-426, 2007.

[107] T. Serre, L. Wolf, and T. Poggio, "Object Recognition with Features Inspired by Visual Cortex," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 994-1000, 2005.

[108] R. P. Würtz, and T. Lourens, "Corner Detection in Color Images through a Multiscale Combination of End-Stopped Cortical Cells," *Image and Vision Computing,* vol. 18, no. 6-7, pp. 531-541, 2000.

[109] L. Itti, and C. Koch, "Computational Modeling of Visual Attention," *Nature Reviews Neuroscience,* vol. 2, no. 3, pp. 194-203, 2001.

[110] C. Koch, and S. Ullman, "Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry," *Human Neurobiology,* vol. 4, no. 4, pp. 219-27, 1985.

[111] P. Montesinos, V. Gouet, and R. Deriche, "Differential Invariants for Color Images," *Proc. International Conference on Pattern Recognition*, pp. 838-840, 1998.

[112] J. Van De Weijer, T. Gevers, and A. D. Bagdanov, "Boosting Color Saliency in Image Feature Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 1, pp. 150-156, 2006.

[113] N. Sebe, T. Gevers, J. Van De Weijer, and S. Dijkstra, "Corner Detectors for Affine Invariant Salient Regions: Is Color Important?,"

Proc. International Conference on Image and Video Retrieval, pp. 61-71, 2006.

[114] A. Guiducci, "Corner Characterization by Differential Geometry Techniques," *Pattern Recognition Letters,* vol. 8, no. 5, pp. 311-318, 1988.

[115] R. Deriche, and G. Giraudon, "Accurate Corner Detection: An Analytical Study," *Proc. International Conference on Computer Vision*, pp. 66-70, 1990.

[116] E. Davies, "Application of the Generalised Hough Transform to Corner Detection," *IEE Proceedings-Computers and Digital Techniques,* vol. 135, no. 1, pp. 49-54, 1988.

[117] R. Deriche, and G. Giraudon, "A Computational Approach for Corner and Vertex Detection," *International Journal of Computer Vision,* vol. 10, no. 2, pp. 101-124, 1993.

[118] P. Brand, and R. Mohr, "Accuracy in Image Measure," *Proc. SPIE Conference on Videometrics III*, pp. 218-228, Boston, USA, 1994.

[119] B. Manjunath, C. Shekhar, and R. Chellappa, "A New Approach to Image Feature Detection with Applications," *Pattern Recognition,* vol. 29, no. 4, pp. 627-640, 1996.

[120] B. Luo, A. D. J. Cross, and E. R. Hancock, "Corner Detection via Topographic Analysis of Vector-Potential," *Pattern Recognition Letters,* vol. 20, no. 6, pp. 635-650, 1999.

[121] F. Shen, and H. Wang, "Corner Detection based on Modified Hough Transform," *Pattern Recognition Letters,* vol. 23, no. 8, pp. 1039-1049, 2002.

[122] T. Tuytelaars, and L. V. Gool, "Matching Widely Separated Views Based on Affine Invariant Regions," *International Journal of Computer Vision,* vol. 59, no. 1, pp. 61-85, 2004.

[123] F. Feng, and T. Pavlidis, "Finding 'Vertices' in a Picture," *Computer Graphics and Image Processing,* vol. 2, pp. 103-117, 1973.

[124] H. Y. F. Feng, and T. Pavlidis, "Decomposition of Polygons into Simpler Components: Feature Generation for Syntactic Pattern Recognition," *IEEE Transactions on Computers,* vol. C-24, no. 6, pp. 636-650, 1975.

[125] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation using Expectation-Maximization and its Application to Image Querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 8, pp. 1026-1038, 2002.

[126] J. J. Corso, and G. D. Hager, "Coherent Regions for Concise and Stable Image Description," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 184-190, 2005.

[127] T. Cour, and J. Shi, "Recognizing Objects by Piecing Together the Segmentation Puzzle," *Proc. Conference on Computer Vision and Pattern Recognition*, 2007.

[128] M. Donoser, and H. Bischof, "Efficient Maximally Stable Extremal Region (MSER) Tracking," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 553-560, 2006.

[129] M. Perdoch, J. Matas, and S. Obdrzalek, "Stable Affine Frames on Isophotes," *Proc. The 11th IEEE International Conference on Computer Vision*, Rio deJaneiro, Brazil, 2007.

[130] J. Bala, K. D. Jong, J. Huang, H. Vafaie, and H. Wechsler, "Using Learning to Facilitate the Evolution of Features for Recognizing Visual Concepts," *Evolutionary Computation,* vol. 4, no. 3, pp. 297-311, 1996.

[131] L. Trujillo, and G. Olague, "Scale Invariance for Evolved Interest Operators," *Proc. 9th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing*, Valencia, Spain, 2007.

[132] P. Viola, and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pp. 511-518, 2001.

[133] A. Neubeck, and L. Van Gool, "Efficient Non-Maximum Suppression," *Proc. 18th International Conference on Pattern Recognition (ICPR)*, pp. 850-855, Hong Kong, 2006.

[134] M. Brown, and D. Lowe, "Invariant Features from Interest Point Groups," *Proc. 13th British Machine Vision Conference*, UK, 2002.

[135] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, 1986.

[136] J. Bigün, "A Structure Feature for Some Image Processing Applications based on Spiral Functions," *Computer Vision, Graphics, and Image Processing,* vol. 51, no. 2, pp. 166-194, 1990.

[137] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell, "A Survey of General-Purpose Computation on Graphics Hardware," *Computer Graphics Forum,* vol. 26, pp. 80-113, 2007.

[138] S. Se, H. K. Ng, P. Jasiobedzki, and T. J. Moyung, "Vision based Modeling and Localization for Planetary Exploration Rovers," *Proc. The 55th International Astronautical Congress*, Vancouver, Canada, 2004.

[139] S. Se, and P. Jasiobedzki, "Stereo-Vision based 3D Modeling and Localization for Unmanned Vehicles," *International Journal of Intelligent Control and Systems,* vol. 13, no. 1, pp. 47-58, 2008.

[140] N. Pettersson, and L. Petersson, "Online Stereo Calibration using FPGAs," *Proc. IEEE Intelligent Vehicles Symposium*, USA, 2005.

[141] C. Cabani, and W. J. MacLean, "A Proposed Pipelined-Architecture for FPGA-based Affine-invariant Feature Detectors," *Proc. Computer Vision and Pattern Recognition Workshop*, 2006.

[142] C. Cabani, "Implementation of an Affine-Invariant Feature Detector in Field-Programmable Gate Arrays," Master of Applied Science Thesis, University of Toronto, 2006.

[143] H. D. Chati, F. Muhlbauer, T. Braun, C. Bobda, and K. Berns, "Hardware/Software Co-Design of a Key Point Detector on FPGA," *Proc. International Symposium on Field-Programmable Custom Computing Machines*, pp. 355-356, 2007.

[144] H. Chati, F. Muhlbauer, T. Braun, C. Bobda, and K. Berns, "SOPC Architecture for a Key Point Detector," *Proc. International Conference on Field Programmable Logic and Applications*, pp. 710-713, 2007.

[145] F. Kristensen, and W. J. MacLean, "Real-Time Extraction of Maximally Stable Extremal Regions on an FPGA," *Proc. IEEE International Symposium on Circuits and Systems*, pp. 165-168, 2007.

[146] D. Kim, K. Kim, J. Y. Kim, S. Lee, and H. J. Yoo, "An 81.6 GOPS Object Recognition Processor based on NoC and Visual Image Processing Memory," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 443-446, 2007.

[147] V. Bonato, E. Marques, and G. A. Constantinides, "A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 18, no. 12, pp. 1703-1712, 2008.

[148] J. Svab, T. Krajnik, J. Faigl, and L. Preucil, "FPGA based Speeded Up Robust Features," *Proc. IEEE International Conference on Technologies for Practical Robot Applications*, pp. 35-41, 2009.

[149] Dimitris Bouris, Antonis Nikitakis, and I. Papaefstathiou, "Fast and Efficient FPGA-Based Feature Detection Employing the SURF Algorithm," *Proc. The 18th IEEE Annual International Symposium*

on Field-Programmable Custom Computing Machines, pp. 3-10, North Carolina, USA 2010.

[150] M. Schaeferling, and G. Kiefer, "Flex-SURF: A Flexible Architecture for FPGA-based Robust Feature Extraction for Optical Tracking Systems," *Proc. International Conference on Reconfigurable Computing and FPGAs*, pp. 458-463, 2010.

[151] M. Schaeferling, and G. Kiefer, "Object Recognition on a Chip: A Complete SURF-Based System on a Single FPGA," *Proc. International Conference on Reconfigurable Computing and FPGAs*, pp. 49-54, 2011.

[152] C. H. Teh, and R. T. Chin, "On the Detection of Dominant Points on Digital Curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 8, pp. 859-872, 1989.

[153] C. Coelho, A. Heller, J. L. Mundy, D. A. Forsyth, and A. Zisserman, "An Experimental Evaluation of Projective Invariants," *Proc. DARPA-ESPRIT Workshop on Applications of Invariants in Computer Vision*, pp. 273-293, Iceland, 1991.

[154] K. Rohr, "Localization Properties of Direct Corner Detectors," *Journal of Mathematical Imaging and Vision,* vol. 4, no. 2, pp. 139-150, 1994.

[155] A. Heyden, and K. Rohr, "Evaluation of Corner Extraction Schemes using Invariance Methods," *Proc. The 13th International Conference on Pattern Recognition*, pp. 895-899, Vienna, Austria, 1996.

[156] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, no. 12, pp. 1338-1359, 1997.

[157] D. Demigny, and T. Kamlé, "A Discrete Expression of Canny's Criteria for Step Edge Detector Performances Evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, no. 11, pp. 1199-1211, 1997.

[158] M. C. Shin, D. Goldgof, and K. W. Bowyer, "An Objective Comparison Methodology of Edge Detection Algorithms using a Structure from Motion Task," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 190-195, Santa Barbara, California, USA, 1998.

[159] M. C. Shin, D. Goldgof, and K. W. Bowyer, "Comparison of Edge Detectors using an Object Recognition Task," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 360-365, Colorado, USA, 1999.

[160] K. Bowyer, C. Kranenburg, and S. Dougherty, "Edge Detector Evaluation using Empirical ROC Curves," *Proc. Conference on*

Computer Vision and Pattern Recognition, pp. 354-359, Colorado, USA, 1999.

[161] S. Baker, and S. K. Nayar, "Global Measures of Coherence for Edge Detector Evaluation," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 373-379, Colorado, USA, 1999.

[162] A. M. Lopez, F. Lumbreras, J. Serrat, and J. J. Villanueva, "Evaluation of Methods for Ridge and Valley Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 21, no. 4, pp. 327-335, 1999.

[163] N. Sebe, Q. Tian, E. Loupias, M. Lew, and T. Huang, "Evaluation of Salient Point Techniques," *Image and Vision Computing,* vol. 21, no. 13-14, pp. 1087-1095, 2003.

[164] F. Mohanna, and F. Mokhtarian, "Performance Evaluation of Corner Detection Algorithms under Similarity and Affine Transforms," *Proc. The British Machine Vision Conference*, pp. 353-362, 2001.

[165] F. Mokhtarian, and F. Mohanna, "Performance Evaluation of Corner Detectors using Consistency and Accuracy Measures," *Computer Vision and Image Understanding,* vol. 102, no. 1, pp. 81-94, 2006.

[166] P. Tissainayagam, and D. Suter, "Assessing the Performance of Corner Detectors for Point Feature Tracking Applications," *Image and Vision Computing,* vol. 22, no. 8, pp. 663-679, 2004.

[167] F. Fraundorfer, and H. Bischof, "Evaluation of Local Detectors on Non-Planar Scenes," *Proc. Austrian Association for Pattern Recognition Workshop*, pp. 125-132, 2004.

[168] F. Fraundorfer, and H. Bischof, "A Novel Performance Evaluation Method of Local Detectors on Non-Planar Scenes," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, USA, 2005.

[169] P. Moreels, and P. Perona, "Evaluation of Features Detectors and Descriptors Based on 3D Objects," *Proc. International Conference on Computer Vision*, pp. 800-807, 2005.

[170] P. Moreels, and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *International Journal of Computer Vision,* vol. 73, no. 3, pp. 263-284, 2007.

[171] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local Features for Object Class Recognition," *Proc. International Conference on Computer Vision*, pp. 1792-1799, 2005.

[172] M. Stark, and B. Schiele, "How Good are Local Features for Classes of Geometric Objects," *Proc. International Conference on Computer Vision*, 2007.

[173] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele, "An Evaluation of Local Shape-based Features for Pedestrian Detection," *Proc. The British Machine Vision Conference*, pp. 11-20, 2005.

[174] A. Haja, B. Jahne, and S. Abraham, "Localization Accuracy of Region Detectors," *Proc.*, IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[175] B. Zeisl, P. Georgel, F. Schweiger, E. Steinbach, N. Navab, and G. Munich, "Estimation of Location Uncertainty for Scale Invariant Feature Points," *Proc. The British Machine Vision Conference*, London, UK, 2009.

[176] W. Förstner, T. Dickscheid, and F. Schindler, "On the Completeness of Coding with Image Features," *Proc. The British Machine Vision Conference*, London, UK, 2009.

[177] H. Aanæs, A. L. Dahl, and K. S. Pedersen, "Interesting Interest Points-A Comparative Study of Interest Point Performance on a Unique Data Set," *International Journal of Computer Vision*, June 2011.

[178] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, "A Comparative Evaluation of Interest Point Detectors and Local Descriptors for Visual SLAM " *Machine Vision and Applications,* vol. 21, no. 6, pp. 905-920, 2010.

[179] M. Asbach, P. Hosten, and M. Unger, "An Evaluation of Local Features for Face Detection and Localization," *Proc. Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 32-35, 2008.

[180] K. Mikolajczyk, and C. Schmid, "A Performance Evaluation of Local Descriptors," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 257-263, Madison, Wisconsin, USA, 2003.

[181] S. Ehsan, A. F. Clark, and K. McDonald-Maier. "JPEG Image Database," http://vase.essex.ac.uk/datasets/index.html.

[182] S. Ehsan, A. F. Clark, and K. McDonald-Maier. "Blur Image Database," http://vase.essex.ac.uk/datasets/index.html.

[183] S. Ehsan, A. F. Clark, and K. McDonald-Maier. "Light Image Database," http://vase.essex.ac.uk/datasets/index.html.

[184] E. Nowak, F. Jurie, and B. Triggs, "Sampling Strategies for Bag-of-Features Image Classification," *Proc. The 9th European Conference on Computer Vision* pp. 490-503, Graz, Austria, May 2006.

[185] T. Tuytelaars, "Dense Interest Points," *Proc. The 23rd IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2281-2288, San Francisco, USA, June 2010.

[186] S. Ehsan, N. Kanwal, A. Clark, and K. McDonald-Maier, "Improved Repeatability Measures for Evaluating Performance of Feature Detectors," *Electronics Letters,* vol. 46, no. 14, pp. 998-1000, 2010.

[187] M. Lillholm, M. Nielsen, and L. D. Griffin, "Feature-based Image Analysis," *International Journal of Computer Vision,* vol. 52, no. 2-3, pp. 73-95, 2003.

[188] S. Lazebnik, C. Schmid, and J. Ponce, "Sparse Texture Representation using Affine-Invariant Regions," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 319–324, Wisconsin, USA, June 2003.

[189] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple Object Class Detection with a Generative Model," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pp. 26-36, New York, USA, 2006.

[190] R. Dragon, M. Shoaib, B. Rosenhahn, and J. Ostermann, "NF-Features–No-Feature-Features for Representing Non-textured Regions," *Proc. ECCV 2010*, pp. 128-141, 2010.

[191] B. Zhang, M. Hsu, and U. Dayal, "K-Harmonic Means-A Spatial Clustering Algorithm with Boosting," vol. LNCS 2007, pp. 31-45, 2001.

[192] S. Ehsan, N. Kanwal, and A. Clark. "Image Database for Coverage based Performance Evaluation," http://vase.essex.ac.uk/datasets/index.html.

[193] J. Neter, W. Wasserman, and G. Whitmore, *Applied Statistics*, Fourth ed.: Allyn and Bacon, 1993.

[194] H. Goldstein, and M. Healy, "The Graphical Presentation of a Collection of Means," *Journal of the Royal Statistical Society-Series A (Statistics in Society),* vol. 158, no. 1, pp. 175-177, 1995.

[195] R. Wolfe, and J. Hanley, "If We're so Different, Why do We keep Overlapping? When 1 plus 1 doesn't Make 2," *Canadian Medical Association Journal,* vol. 166, no. 1, pp. 65-66, 2002.

[196] D. E. Wrede, "Central Axis Tissue-Air Ratios as a Function of Area/Perimeter at Depth and their Applicability to Irregularly Shaped Fields," *Physics in Medicine and Biology,* vol. 17, no. 4, pp. 548-554, 1972.

[197] S. Ehsan, A. F. Clark, and K. McDonald-Maier. "Database for Image Registration task," http://vase.essex.ac.uk/datasets/index.html.

[198] W. Cheung, and G. Hamarneh, "n-SIFT: n-Dimensional Scale Invariant Feature Transform," *IEEE Transactions on Image Processing,* vol. 18, no. 9, pp. 2012-2021, 2009.

[199] A. Abdel-Hakim, and A. Farag, "CSIFT: A SIFT Descriptor with Color Invariant Characteristics," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1978-1983, USA, 2006.

[200] K. Mikolajczyk, "Detection of Local Features Invariant to Affine Transformations Application to Matching and Recognition," PhD Thesis, Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII), Institut National Polytechnique de Grenoble (INPG), France, 2002.

[201] N. Zhang, "Computing Parallel Speeded-Up Robust Features (P-SURF) via POSIX Threads," *Lecture Notes in Computer Science,* vol. 5754, pp. 287-296, Sep. 2009.

[202] H. Bay, "From Wide-baseline Point and and Line Correspondences to 3D," Doctor of Sciences Thesis, Swiss Federal Institute of Technology, ETH Zurich, 2006.

[203] H. Bay, B. Fasel, and L. Van Gool, "Interactive Museum Guide: Fast and Robust Recognition of Museum Objects," *Proc. First International Workshop on Mobile Vision*, May 2006.

[204] P. Cattin, H. Bay, and L. Van Gool, "Retina Mosaicing using Local Features," *Proc. MICCAI*, 2006.

[205] S. Lee, Y. Zhang, Z. Fang, S. Srinivasan, R. Iyer, and D. Newell, "Accelerating Mobile Augmented Reality on a Handheld Platform," *Proc. 27th IEEE International Conference on Computer Design*, pp. 419-426, California, USA, October 2009.

[206] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-Local Affine Parts for Object Recognition," *Proc. 15th British Machine Vision Conference*, UK, 2004.

[207] H. Bay. "Original SURF Code," http://www.vision.ee.ethz.ch/~surf/download_ac.html.

[208] N. Cornelis, and L. Van Gool, "Fast Scale Invariant Feature Detection and Matching on Programmable Graphics Hardware," *Computer Vision and Pattern Recognition (CVPR) Workshop*, June 2008.

[209] "OpenCV Library," http://sourceforge.net/projects/opencvlibrary.

[210] S. Srinivasan, Z. Fang, R. Iyer, S. Zhang, M. Espig, D. Newell, D. Cermak, Y. Wu, I. Kozintsev, and H. Haussecker, "Performance Characterization and Optimization of Mobile Augmented Reality on

Handheld Platforms," *Proc. IEEE International Symposium on Workload Characterization*, pp. 128-137, USA, October 2009.

[211] C. Evans. "Open SURF," http://code.google.com/p/opensurf1/.

[212] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874, 2006.

[213] H. Jegou. "Copydays Data Set," http://lear.inrialpes.fr/~jegou/data.php.

[214] J. van de Weijer. "Blur Data Set," http://lear.inrialpes.fr/people/vandeweijer/data.

[215] M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid, "Evaluation of GIST descriptors for Web-Scale Image Search," *Proc. CIVR*, 2009.

[216] J. van de Weijer, and C. Schmid, "Blur Robust and Color Constant Image Description," *Proc. IEEE International Conference on Image Processing*, pp. 993-996, Atlanta, USA, 2006.

[217] F. Crow, "Summed-area tables for texture mapping," *ACM SIGGRAPH Computer Graphics,* vol. 18, no. 3, pp. 212, 1984.

[218] M. Grabner, H. Grabner, and H. Bischof, "Fast Approximated SIFT," *Proc. Asian Conference on Computer Vision*, pp. 918-927, 2006.

[219] B. Kisacanin, "Integral Image Optimizations for Embedded Vision Applications," *Proc. IEEE SouthWest Symposium on Image Analysis and Interpretation*, pp. 181-184, 2008.

[220] C. Messom, and A. Barczak, "Stream Processing of Integral Images for Real-Time Object Detection," *Proc. Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 405-412, 2008.

[221] H. C. Lai, M. Savvides, and T. Chen, "Proposed FPGA Hardware Architecture for High Frame Rate (>100 fps) Face Detection using Feature Cascade Classifiers," *Proc. First IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2007.

[222] N. Zhang, "A Novel Parallel Prefix Sum Algorithm and its Implementation on Multi-Core Platforms," *Proc. Second International Conference on Computer Engineering and Technology*, pp. 66-70, 2010.

[223] M. Hiromoto, K. Nakahara, H. Sugano, Y. Nakamura, and R. Miyamoto, "A Specialized Processor Suitable for AdaBoost-based Detection with Haar-like Features," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, USA, 2007.

[224] C. Gao, and S. L. Lu, "Novel FPGA based Haar Classifier Face Detection Algorithm Acceleration," *Proc. International Conference on Field Programmable Logic and Applications*, pp. 373-378, Germany, 2008.

[225] Y. Wei, X. Bing, and C. Chareonsak, "FPGA Implementation of AdaBoost Algorithm for Detection of Face Biometrics," *Proc. IEEE International Workshop on Biomedical Circuits and Systems* 2004.

[226] T. Theocharides, N. Vijaykrishnan, and M. J. Irwin, "A Parallel Architecture for Hardware Face Detection," *Proc. IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, 2006.

[227] N. Zhang, "Working towards Efficient Parallel Computing of Integral Images on Multi-Core Processors," *Proc. Second International Conference on Computer Engineering and Technology*, pp. 30-34, Chengdu, China, 2010.

[228] J. Hensley, T. Scheuermann, M. Singh, and A. Lastra, "Interactive Summed-Area Table Generation for Glossy Environmental Reflections," *Proc. ACM SIGGRAPH*, 2005.

[229] J. Cho, S. Mirzaei, J. Oberg, and R. Kastner, "FPGA-based Face Detection System using Haar Classifiers," *Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays,* pp. 103-112, Monterey, California, USA, 2009.

[230] J. Hensley, T. Scheuermann, G. Coombe, M. Singh, and A. Lastra, "Fast Summed-Area Table Generation and its Applications," *Computer Graphics Forum,* vol. 24, no. 3, pp. 547-555, 2005.

[231] M. Yang, Y. Wu, J. Crenshaw, B. Augustine, and R. Mareachen, "Face Detection for Automatic Exposure Control in Handheld Camera," *Proc. Fourth IEEE International Conference on Computer Vision Systems*, 2006.

[232] H. C. Lai, R. Marculescu, M. Savvides, and T. Chen, "Communication-Aware Face Detection using NOC Architecture," *Proc. Sixth International Conference on Computer Vision Systems*, pp. 181-189, 2008.

[233] V. Nair, P. O. Laprise, and J. J. Clark, "An FPGA-based People Detection System," *EURASIP Journal on Applied Signal Processing,* vol. 2005, no. 7, pp. 1047-1061, 2005.

[234] H. T. Ngo, R. C. Tompkins, J. Foytik, and V. K. Asari, "An Area Efficient Modular Architecture for Real-Time Detection of Multiple Faces in Video Stream," *Proc. Sixth International Conference on Information, Communications and Signal Processing*, 2007.

[235] S. Sengupta, A. E. Lefohn, and J. D. Owens, "A Work-Efficient Step-Efficient Prefix Sum Algorithm," *Proc. Workshop on Edge Computing Using New Commodity Architectures*, 2006.

[236] G. E. Blelloch, "Prefix Sums and their Applications," *Synthesis of Parallel Algorithms*, J. H. Reif, ed., pp. 35-60, 1990.

[237] D. Horn, "Stream Reduction Operations for GPGPU Applications," *GPU Gems*, M. Pharr, ed., pp. 573-589: Addison Wesley, 2005.

[238] Y. Sato, T. Sugimura, H. Noda, Y. Okuno, K. Arimoto, and T. Nagasaki, "Integral-Image based Implementation of U-SURF Algorithm for Embedded Super Parallel Processor," *Proc. International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 485-488, 2009.

[239] B. Bilgic, B. K. P. Horn, and I. Masaki, "Efficient Integral Image Computation on the GPU," *Proc. IEEE Intelligent Vehicles Symposium*, pp. 528-533, San Diego, USA, 2010.

[240] A. Juan, "Field-Programmable Gate Array Implementation of a Scalable Integral Image Architecture Based on Systolic Arrays," Master of Science Thesis, Utah State University, 2011.

[241] Y. T. Wu, C. Y. Cho, S. Y. Tseng, C. N. Liu, and C. T. King, "Parallel Integral Image Generation Algorithm on Multi-Core System," *Proc. 9th IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 31-35, 2011.

[242] T. B. Terriberry, L. M. French, and J. Helmsen, "GPU Accelerating Speeded-Up Robust Features," *Proc. 3DPVT*, pp. 355-362, 2008.

[243] H. J. W. Belt, "Word Length Reduction for the Integral Image," *Proc. The 15th IEEE International Conference on Image Processing*, pp. 805-808, San Diego, California, USA, 2008.