Experimental Exploration of Interest Point Repeatability for 3D Objects and Scenes

by

Simon Richard Lang, *B.IT. (Hons)*
School of Informatics and Engineering,
Faculty of Science and Engineering

August 13, 2020

A thesis presented to the
Flinders University of South Australia
in total fulfillment of the requirements for the degree of
Doctor of Philosophy

Adelaide, South Australia, 2020

Contents

List of Figures

List of Tables

Abstract

In Computer Vision, finding simple features often entails filtering a 2D image to find basic patterns. Often these are used for tracking or co-ordination (e.g. in stereo vision, or the motion vectors enabling MPEG compression), and requires a method for measuring tracking performance, usually through repeatability of simple points of interest.

Interest points are used regularly in Computer Vision in various applications. However, those applications, along with their requirements, are becoming increasingly complex and demanding. 2D interest point detectors (such as the Harris detector) are now regularly applied to 3D environments with varying degrees of success. With forays into automated optimisation of 2D feature classifiers via GP, there is a demand for better evaluation approaches that are more relevant to 3D, but the disconnect between 2D and 3D means they are still tightly coupled to their respective domains. 2D interest points are, by their design, poorly optimised for 3D, or even unoptimisable, due to their inability to work with, or awareness of, scene depth. Currently, evaluation of 2D-based interest points has innovated little in recent years, with research still highly dependent on image datasets and approximations of 3D, and little, or no reliable ground truth.

Interestingly some approaches prove effective even though not specifically designed to find interest points, notably the Fast and Harris detectors. Since these are effective in 2D images of 3D scenes, it seems they must be capturing some kind of 3D information, but this raises the question of how optimal they are. Testing 2D-based detectors in a real-world environment is not normally possible though due to their inability to be properly assessed, but a virtual scene can help bridge this gap. The concept of virtual spaces are seeing greater usage as they can address problems where a real world environment can't be used. Through the use of a virtualised ground truth, it is possible to probe aspects of the real world to solve problems or gain insight, where in normal situations, it would not be possible.

This thesis seeks to utilise virtual 3D spaces to bridge the gap between 2D detectors and 3D scenes, by emulating performance evaluation of 2D interest points with virtual spaces. A virtual ground truth can evaluate features detected based on 2D interest points in a simulated 3D space, for realistic evaluation of repeatability performance. By doing so, a virtual scene is able to utilise more

sophisticated evaluation strategies like ROC, and informedness, for correct feature classification. This enables 2D feature detectors to be properly evaluated, and potentially optimised, when tested for real world applications in virtual 3D spaces.

To test these new approaches, a virtual space is used to emulate well known repeatability evaluation via 2D and Euclidean space to find closest points. This is tested using conventional 2D detectors, as well as GP-based optimisation of 2D classifiers. Both images sets, and 3D-scanned model datasets are used in testing, as well as comparing the performance of 2D, 3D, and color optimisation approaches with GP. Additionally, we demonstrate that the use of virtual spaces enables other types of evaluation approaches like “informedness” which can similarly evaluate performance based on ϵ thresholds. Informedness demonstrates that it can incorporate more information about classified features identified in 2D, and can more effectively evaluate classifier performance due to the virtual ground truth. Our tests also empirically support that optimisation with depth data not only optimised 2D classifiers in virtual spaces comparatively better than without it, but also supports the argument that certain well accepted conventions regarding interest point repeatability and the Moore neighbourhood ($\epsilon = 1.5$) are not necessarily the best best performance tradeoff.

Certification

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

As requested under Clause 14 of Appendix D of the *Flinders University Research Higher Degree Student Information Manual* I hereby agree to waive the conditions referred to in Clause 13(b) and (c), and thus

- Flinders University may lend this thesis to other institutions or individuals for the purpose of scholarly research;
- Flinders University may reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signed Dated 13/08/2020

Simon Richard Lang

Acknowledgements

For me, this thesis has been a series of battles, and every single one of these battles would not have been won without the assistance and support of those around me. It is my strong belief that without their experience, professionalism, and, most vitally, their understanding and consideration of my own personal circumstances, that helped bring this thesis to the finish line. Without them, this war with my own self doubt, misfortune, and the fact that I probably bit off more than I could chew, would never have been won.

My thanks to faculty staff Jennie Brand, Mark Taylor, and Natalie Hill for their unending patience, and understanding that life happens, and enabling me to keep pushing onward, as it was never something I took for granted. My enormous thanks to my principal supervisor Prof. David Powers for always being patient with me and ready to help. I must also acknowledge my co-supervisor, Dr. Martin Luerssen, whose advice served as the backbone of much of the work in this thesis, and I can only guess how many hundreds of hours he sacrificed in weekly meetings over the years to help bring this work to fruition. Your advice, occasional debugging skills, and being the sounding board/filter for all the crazy ideas, before I took them to David was invaluable. Your patience with me as I bumbled through my thesis made all the difference.

Professional editor, Joe Miller at editbureau.com.au, provided copyediting and proofreading services, according to the guidelines laid out in the university-endorsed national Guidelines for editing research theses. Joe's advice and professionalism during editing was greatly appreciated.

Acronyms

This is a list of all acronyms that appear in this document. They are listed in the order in which they first appear.

CV	Computer Vision
EC	Evolutionary Computing
EA	Evolutionary Algorithm
GA	Genetic Algorithm
ES	Evolution Strategies
EP	Evolutionary Programming
GP	Genetic Programming
IP	Interest Point
STEIPR	System for Testing Evolved Interest Point Repeatability

Chapter 1

Introduction

1.1 Introduction

This chapter is an introduction to the thesis and outlines the motivations, goals, contributions and overview of the thesis as a whole. A brief background is provided first 1.2, followed by an outline of the research goals, which serves to establish the focus of this dissertation 1.3, along with a clarification of its intent. The chapter finishes with a list of the major contributions of the thesis 1.4, and an outline of the thesis' organisation 1.5.

1.2 Motivation

Performance evaluation in the field of computer vision is a topic that underpins much of the field and forms the basis of many of its branches. In the last few decades especially, there have been a number of new approaches to computer vision that demand new forms of evaluation so that the benefits of these approaches can be properly quantified and compared. Image analysis requires metrics that are applicable to the features being measured, and relevant to the application domain. An interest point is typically characterised as being mathematically well defined, with a well defined position in an image, information rich, stable, and preferably robust towards scale changes (Lindeberg 2013, Lindeberg 2015*a*). Interest point detection in computer vision is a process involving detection of differences in the area of an image (Zhang, Cagnoni & Olague 2009) which in most cases, are the pixels of an image. The role, and nature of an interest point can vary greatly, but a few examples of the applications of interest points involve robotics (Beni & Wang 1993) 3D scene reconstruction (Artieda, Sebastian, Campoy, Correa, Mondragón, Martínez & Olivares 2009) and object tracking (Gauglitz, Hllerer & Turk 2011).

Features in interest point detection are the result of a transformation of raw pixel data in an image that can usually be tied to a central position. The feature in question may, or may not be useful depending on the nature of the interest point detector. Image processing is done for myriad purposes, but one underlying reason is for enhancing certain desirable features (eg. such as a feature sharing similar qualities like edges or corners), while suppressing others (eg. random noise, blank space). The techniques used to achieve the desired outcome can range from the simplistic to the dizzyingly complex. Computer vision goes one step further to not simply process a single image, but to become a robust and rigorous feature enhancement/suppression process that can be used on a variety of images, in a variety of conditions, to zero in on very specific features at a low level. This can then be further processed to accurately interpret the contents of a scene, not just in a single image, but also across multiple images. Applications can vary a great deal, from enhancement of video compression, to being used to make intelligent decisions, and can cross the boundary to where intelligent systems can become aware of, and intelligently interact with the environment being perceived. For decades, computer vision researchers have been constantly striving to create more sophisticated and robust computer vision algorithms (Abend, Harley & Kanal 1965, Ullmann & Kidd 1969, Haralick, Shanmugam, Dinstein et al. 1973) such that eventually, designers will no longer need to point out to intelligent systems which features they need to look for; such systems will instead assess and refine themselves unassisted (LeCun, Boser, Denker, Henderson, Howard, Hubbard & Jackel 1989, Sebastiani 2002, Bandyopadhyay & Maulik 2002, Cheng, Cai, Chen, Hu & Lou 2003, Omran, Engelbrecht & Salman 2004, Hinton, Osindero & Teh 2006, Trujillo & Olague 2008, Le 2013, Goodfellow, Bengio, Courville & Bengio 2016).

When it comes to robotics and other fields (Bhanu, Lin & Krawiec 2005) that focus on features and image processing, this is the ultimate goal. To find a computer vision system that can operate with little, to no assistance would save tremendous amounts of time and resources, as well as opening new pathways for innovation, and as inspiration when considering new computer vision algorithms. Though image classification has advanced substantially over the years, its fundamentals have not changed greatly. Particularly with simple 2D features, like corners, edges, and blobs, the field has established techniques for feature identification, extraction, and evaluation (Schmid, Mohr & Bauckhage 2000) and though these have improved to adapt to the manner in which features are classified (Mikolajczyk & Schmid 2005), these methodologies have seen little improvement with regard to 2D feature evaluation in recent times. These techniques for low level feature detection often serve as the basis for higher-level elucidation of a larger feature set, or reconstruction (Johnson & Hebert 1999, Viola & Jones 2001*a*, Lowe 2004, Mozos, Gil, Ballesta & Reinoso 2007).

Research on genetic programming has demonstrated that it is possible to codify existing feature detectors in a manner that enables them to be optimised with evolutionary learning techniques (Ebner, karls-universitat Tubingen & Rechnerarchitektur 1998). This state-of-the-art technique still uses 2D images, and a simulation of 3D is restrictive, inflexible, and is constrained by the detail of the image it possesses, as well as how the images can be processed. The establishment of sufficient ground truth still requires substantial effort, and is imperfect and limiting. These limitations mean that large classes of tests are unavailable and ultimately, untestable using existing evaluation methods. This state of affairs is somewhat backwards considering that older machine learning work sought to utilise other evaluation methodologies that were arguably better (Bradley 1997, Viola & Jones 2004), but are now largely overlooked.

Even in light of the fact that the field of computer vision is increasingly moving towards a more 3D-centric application domain (Guo, Bennamoun, Sohel, Lu, Wan & Kwok 2016*a*, Yi, Trulls, Lepetit & Fua 2016), there is still a constraint on the types of scenes that 2D detectors can evaluate. Optimisation with Genetic Programming (GP) is similar to conventional detectors, and is appropriate for largely static scenes, but many early image detectors were never designed to handle the sophistication of depth, realistic scene deformation, or high accuracy/detail that moderm technology is capable of capturing. Dynamic and complex scenes, especially those that require the triangulation of points in a 3D space are ultimately untestable if a sufficiently reliable and accurate ground truth is not available.

It is argued in this dissertation, that without the assistance of a system that can emulate the complex environments of the real world, an optimised detector can not be effectively created for them. To leverage the opportunities that genetic programming provides, a suitable 3D ground truth needs to be developed to allow the development of detectors that perform better in real world environments compared to the more simplistic 2D abstractions of real world environments that are currently utilised in the field. As will be seen in the literature section of this thesis, there are certain issues that require addressing:

1. Given the fact that interest point detectors have no knowledge of depth (and may not even be designed to take advantage of it), the use of a 3D-based ground truth environment needs to seamlessly emulate existing evaluation methods. If any advantages of a virtualised testing environment are going to be beneficial, they must first be able emulate other evaluation processes that are well utilised. This is critical if it is to serve as a tool for other fields like GP, which depend on such metrics.

2. A 3D ground truth should be able to leverage the depth of a scene and detected points based on the scene to identify precisely where within the 3D environment the feature is positioned.

3. An endeavour should be made to demonstrate aspects of the 3D ground truth during testing that are difficult or infeasible to be tested in a 2D, or real-world environment.

4. Unlike existing approaches that try to build ground truth from highly calibrated real-world images, a 3D-based ground truth needs to be highly accurate and reliable, such that it is able to map features, and be resolution-independent to ensure the highest possible accuracy.

5. Given that the ground truth is 3D, it should be able to infer better performance of features than 2D alone. However, it is unknown whether existing repeatability methods will reflect that performance improvement, so it may be necessary to investigate and develop new evaluation methodologies based on the ground truth within the 3D scene to take advantage of this.

6. In the pursuit of a superior form of ground truth (compared to 2D images, and real-world scenes), it is likely the ground truth model will have limitations in what it can do, or caveats that affect performance evaluation that need to be investigated so they can be addressed or avoided.

1.3 Research Goals

The overarching goal of this thesis is to develop new evaluation approaches that provide better performance measuring compared to existing evaluation approaches, and evaluation of 2D-based detectors via a virtual 3D environment, that can be adapted to existing evaluation approaches in the computer vision field but also perform other beneficial functions that are not possible, or at least costly, normally. In order to achieve this, this thesis addresses the following research objectives.

1. Develop a virtualised 3D space to test existing computer vision algorithm performance with 3D models with the necessary functionality required to duplicate existing performance evaluation of interest points.
2. Use the virtual space to emulate testing conditions that are normally used to test already-existing interest point detectors.
3. Integrate the 3D testing environment with existing GP applications to demonstrate that is capable of performance evaluation and optimisation of interest point detectors unassisted.
4. Demonstrate that the performance of conventional and optimised detectors can be evaluated and compared under varying 2D/3D virtual space conditions in a more comprehensive manner than existing evaluation approaches.
5. Demonstrate that the utilisation of 3D depth can afford a measurable improvement to detector performance when compared to the absence of depth.
6. Develop other analytic strategies based on the 3D ground truth that are able to provide better analysis of performance (and under certain conditions improved performance) compared to techniques that do not utilise scene depth.

1.4 Major Contributions

This thesis explores the capabilities of a virtualised 3D-based ground truth for the testing of existing, and evolved interest point detectors, in conjunction with 3D models and existing performance evaluation methodologies that are examined in the literature. For the evaluation of GP-based interest point detectors with the assistance of scene depth, there is no prior domain knowledge available. The major contributions of the thesis are as follows:

1. It shows how a virtualised 3D-based ground truth environment can be integrated with 2D-based interest point detectors, and image descriptors to

process the scene for features. The thesis demonstrates that such a 3D-based ground truth environment is capable of deforming/transforming the scene to emulate changes in camera viewpoint, noise, lighting, rotation and scale. It also demonstrates that it is able to emulate the same performance evaluations mentioned in the literature to gauge repeatability with a higher-than-normal degree of accuracy. This is demonstrated with standard 2D images from the BSD dataset (Martin, Fowlkes, Tal & Malik 2001), and a range of 3D scanned (*Stanford 3D Scanning Repository* n.d.) and other various models.

2. It demonstrates that in addition to testing keypoint repeatability using conventional metrics that are well established in the Computer Vision (CV) field, a virtualised 3D-based ground truth environment can be used to emulate other testing methodologies utilised in the field of GP, by being able to train interest point operators with the use of normal 2D image datasets, as well as with the use of 3D scanned models under various conditions. This utilisation of virtual ground truth integrates seamlessly with existing research described in the literature, and highlights that it can completely replace 2D image datasets, with little need for time-intensive dataset preparation.

3. It also re-examines the use of colour, which builds on previous work done in the GP area (Shao, Liu & Li 2014), but utilises the virtual ground truth of 3D scenes instead.

4. It highlights existing deficits in existing repeatability evaluation, and that the improved ground truth can be replaced with a more rigorous evaluation metric called *Informedness*, which is only made possible by the use of the improved 3D ground truth. This highlights that existing repeatability evaluation is potentially misleading for classification of features under 2D conditions, compared to repeatability that utilises 3D interest point data.

5. Informedness also shows that it can better analyse the repeatability performance of detectors by seamlessly incorporating well-established and accepted threshold metrics in CV. This provides an extremely simplified means of comparing the effects of Informedness performance to existing repeatability metrics at each error threshold as well as illustrating the best cost/benefit tradeoff at each threshold.

6. The above technique enables the performance of a 2D detector to be analysed in a 3D environment in isolation without needing to know, or utilise, 3D data or scene information, but still illustrate a cost/benefit for each error

threshold. It also demonstrates, that the utilisation of a single defacto threshold (which is $\epsilon = 1.5$ in CV) is not necessarily ideal.

7. Unlike existing repeatability metrics, the virtual ground truth, in conjunction with the ability to measure 2D and 3D repeatability of points, can optimise 2D detectors to perform better with the assistance of 3D interest point repeatability compared to 2D alone. This optimisation improvement is an emergent, and passive effect of the utilisation of 3D data, and is empirically shown to occur even with utilisation of fitness functions that are older, and (arguably) supersceded by newer approaches. These effects are only possible to gauge and discern by using a 3D-based ground truth in conjunction with Informedness as it is undetectable, or even misleading with existing repeatability measures.

1.5 Overview

This thesis begins with a review of the main concepts and literature that underpin the research contained herein. The review covers the two main branches of research separately, then explains how both are merged to achieve the main contributions of this thesis. The latter half of the review covers the main research goals via a broad technical overview, followed by experimental testing and evaluation, which builds on previous experiments, and an analysis of the results, which forms the basis for further exploration. The chapters are outlined as follows.

Chapter 2 briefly covers the application of evolution in computer science, and outlines the distinctive differences that sets genetic programming apart. It also covers the basic underlying principals of genetic programming, and its application in the field of computer science in general. This chapter serves a basic introduction to the Evolutionary Learning principals that will be covered in more detail in subsequent chapters. The chapter ends with a brief summary of cases where Evolutionary Learning techniques have been applied to problems in Computer Vision research.

Chapter 3 describes an approach to effectively emulate existing repeatability via the use of a 3D virtual space. This is tested via simple affine transforms of images, as well as 3D models to demonstrate it can sufficiently measure performance. A variety of tests are performed using rotation, scale, and noise to assess the performance of existing interest point detectors.

Chapter 4 presents a novel process for more precisely measuring detector repeatability via Informedness thanks to the more reliable ground truth provided for by the virtual space. Through empirical testing, it serves to establish that it can reliably measure the performance of detectors using well accepted testing metrics in Computer Vision and demonstrates that it is able to empirically argue that existing conventions such as $\epsilon = 1.5$ are not necessarily ideal as the defacto threshold when assessing repeatability. A variety of well known interest point detectors are tested under well established testing conditions and their performance compared.

Chapter 5 details the integration of the previous chapter's work with GP to create classifiers that are optimised using 2D and 3D data of 2D keypoints. This establishes a fully functional framework that can evolve interest point detectors with a completely virtual 3D environment in a controlled manner. Tests are performed on various scene configurations such as simple images, colour/greyscale, and 3D models.

Chapter 6 builds on the success of the previous chapter to test a larger variety of variation to the virtual scene configuration such as max points per scene, lighting, and image resolution in 2D and 3D training configurations.

Chapter 7 outlines the conclusions, main contributions of this dissertation and the results discovered, as well as highlight improvements that could have been made. From this, suggestions for future research are explored such as the use of Informedness as part of the fitness metric to replace Schmid repeatability during GP training, refinement of tests that statistically demonstrated a divergence between Schmid and Informedness performance, expanding the testing to more complex scenes, as well as implementing conventional detectors such that they can be more properly analysed and compared in these environments.

Chapter 2

Literature review

This chapter is split into three major sections. Firstly, a review of the field of features, interest points, and to a more limited degree, key point detectors is undertaken, as well as a particular focus on evaluation of the performance of these detectors. The section will also cover the basic concepts of simple feature detectors, in particular corner and edge detection. The second component of this chapter will introduce a background of evolutionary computation, in particular genetic programming. Lastly will be a review of the overlap of computer vision, and genetic programming, and, to a lesser extent, swarm intelligence. The purpose of the chapter is to establish the current state of interest point detector evaluation, and its utilisation in optimisation of feature detection in the EC field.

2.1 Overview of Keypoints and Features

Interest points and interest point detectors are generalised terms used where a feature classifier has been calibrated to identify specific features in a digital image with semantic meaning. “Feature” as a term can be quite ambiguous, as it relies heaviliy on arbitrary interpretation of what a feature may be. In the most abstract sense, it can be defined as a measurable value of an attribute (Bishop 2006). For example, the angle of a corner could represent an attribute, and 45 degrees the value of that attribute.

When it comes to the discussion of simple features, this dissertation defines them as features that have been identified as being something of interest (as in a position of indeterminate size in an image), but not necessarily useful, and the terms “interest point” and “keypoint” are used interchangeably. Interest points are generally derived from an analysis of neighbouring pixels or region of interest (ROI), also referred to as keypoints. Successful identification of an interest point results in only the location of the point being used in further processing, which typically categorises this type of processing as an interest point (Tuytelaars & Mikolajczyk 2008). Developing feature extraction techniques for these keypoint types has been an ongoing research problem for many years now, and the literature on this subject is immense. Though not a complete list, these are some of the more common keypoints that detectors have been designed to find:

- Edges: These could be straight or curved lines (Moreno, Puig, Julià & Garcia 2009, Papari & Petkov 2011), or more precise definitions like ramp (Petrou & Kittler 1991), stair (Lunscher & Beddoes 1986), or texture edges (Tan 1995).
- Blobs: Large areas of the image are identified as spaces that show a degree of consistency (Forssén 2007).
- Corners: Similar to an edge, but at some point the edge suddenly changes direction. Thus the edge becomes a corner (Harris & Stephens 1988).
- Ridges: Most commonly-identified structures. Such as roads (Eberly, Gardner, Morse, Pizer & Scharlach 1994, Lindeberg 1998).

There are many sub-categories of interest point detectors to deal with different types of features, at varying levels of complexity. Examples of more common techniques include, but are not limited to, the Sobel detector (Sobel & Feldman 1968) , Canny (Canny 1986), Laplacian of Gaussians (LoG) (Tabbone & Lorraine 1993), difference of Gaussians (DOG) (Bundy & Wallen 1984), Harris (Harris & Stephens 1988), MSER (Matas, Chum, Urban & Pajdla 2004), grey-level blob type detectors (Lindeberg 1993) and principal curvature-based region detector (PCBR) (Deng, Zhang, Mortensen, Dietterich & Shapiro 2007).

2.1.1 Characteristics of Features in CV

Tuytelars (2008) defines a feature as a localised spatial region of pixels that is not limited by colour or texture boundaries. Features can overlap each other, and can be identified as points, regions or edge segments. Tuytelars outlines the best properties that a feature should have:

- Repeatability: When presented with two or more images of an object or scene, where the scene appears in both images, the detected features should appear in both images. This can be achieved by invariance, or by robustness. Invariance aims to mathematically analyse image deformations so that any variance between images can be reduced. Robustness focuses on reducing the sensitivity of the detector, which may reduce accuracy, but also helps to avoid false positives. Robustness usually works better on small deformations in the image such as noise, compression artefacts, blur, etc.

- Distinctiveness: Each feature found should contain enough variation so that it can be distinguished from other detected features. This helps to distinguish the feature as a distinct spatial point that can be reliably found again in other images containing the same point.

- Locality: Detected features should be local, to avoid the possibility of occlusion.

- Quantity: The number of features should be numerous enough so that small objects can be distinguished, yet be able to adopt intuitive thresholds that reflect the information contained within the image, and act as a compact image representation.

- Accuracy: Features should be localised with respect to image location and scale.

- Efficiency: A feature detector should, if possible, be fast and efficient when detecting features in an image.

- Quantity: All the existing features in the image should be detected.

Within the scope of these listed criteria, distinctiveness and locality conflict: as one increases, the other's effectiveness is lessened. This is also the case with distinctiveness versus invariance, and distinctiveness versus robustness. In these cases, a reduction in the information contained in a feature makes it difficult for that feature to be repeated across subsequent images. Compounding the situation is the conflict between invariance and robustness, since robustness aims to detect features by reducing the detection accuracy in the image, whereas invariance depends on higher accuracy.

2.1.2 Feature Processing

Processing of features in a classifier has a number of sub classifications, including extraction, construction, and selection of features (Liu & Motoda 1998, Chandrashekar & Sahin 2014), with feature selection being the main focus of this dissertation. Feature extraction, in simplified terms, is the process of taking input data and reducing the information into a salient representation for further processing. This could take the form of a histogram of pixel values, performing a statistical analysis of a range of pixel values such as identification of outliers, finding the mean, or variance, within a region. This can reduce noise (El Ferchichi, Zidi, Laabidi, Ksouri & Maouche 2011), and help isolate candidate features for future processing in computer vision processing (Viola & Jones 2004).

Unlike feature extraction, feature construction operates on a set of pre-extracted features, which in simplified terms act to filter out less relevant features in order to create a more relevant and concentrated pool of features to work with (Neshatian, Zhang & Andreae 2012). As the term “feature selection” implies, its objective is to select relevant features over less relevant ones (Liu & Motoda 1998), to achieve this objective, different approaches can be used, and are categorised as the sub-groups filter, wrapper and embedded (Liu & Motoda 1998). The filter approach to feature selection is performed as part of pre-processing, with the objective of removing data that would not be useful (Koller & Sahami 1996).

In the wrapper-based approach, a searching method is used in conjunction with a classifier which determines the quality of the features (Neshatian et al. 2012). The wrapper-based approach is a computationally expensive process, however, due to the number of evaluations that need to be performed for each feature selected (John, Kohavi & Pfleger 1994). Subsequent cross-validation is necessary as well, to ensure the selected features are robust and provide adequate prediction of the features (Miller 2002). The embedded approach uses its own feature selection algorithm as part of the search algorithm. In other words, there is no distinction between the selection of features, and the learning of features, which enables a more cohesive learning process such as in the case of decision trees (Quinlan 1986) or random forest (Breiman 2001). GP can be represented in both feature construction and selection (Ahmed, Zhang, Peng & Xue 2014), but is mostly considered a strategy for feature selection (Russell & Norvig 2016, Tran, Xue & Zhang 2016, Xue, Zhang, Browne & Yao 2016).

2.1.3 Types of Interest Point Detectors

The following is an overview of the different approaches that have been applied to computer vision over the last few decades. It is by no means all-encompassing, but it aims instead to highlight the diversity of approaches that have been undertaken in the field of computer vision.

2.1.3.1 Contour Curvature Based Methods

These can be based on contour curvature; however, despite their name, they can also be based on other data/methods. Contours are usually a chain of points. Once a chain is identified, it can be analysed for corners (Blake & Isard 1998). An adaptation of this method has been to use deformable models, also referred to as “snakes”, to identify boundaries (McInerney T 1996). Other more recent methods utilise what is termed the Second Order of Difference Contours (SODC) (Lin, Zhu, Zhang, Huang & Liu 2017) , as well as deriving points via direction Gaussian derivatives (Zhang & Shui 2015).

2.1.3.2 Intensity Based Methods

Other methods are based on intensity, where the variance in the values is used to identify features. This is done via first- and second-order grey value derivatives, or through the use of heuristics to find high variance. Although the assumptions made by the detector are broad, their advantage is that they can be applied to a very wide range of images. The Harris detector (Harris & Stephens 1988) is one of the earliest and still one of the more reliable intensity-based interest point detectors. Another variation of the intensity approach is the smallest univalue segment assimilating nucleus (SUSAN) detector (Smith & Brady 1997) which, unlike the Harris detector (which uses image derivatives), uses a mask and matched pixel intensity.

2.1.3.3 Biologically Inspired Methods

As opposed to the other methods described previously, which were designed with a specific application purpose, biologically-derived detection models the behaviour of the brain, in order to derive useful image processing techniques. Biologically-derived methods have also been formulated, which harness the pre-attentive and attentive stages of human-based vision (Neisser 1964). In the pre-attentive stage, distinctive elements within the image are identified. In human vision, these are usually elements that stick out and deserve greater attention than the rest of the scene. During the attentive stage these areas that have been identified are then further processed. This approach to image processing, where initial features are identified, and the features are then grouped, is a widely-adopted approach in computer vision.

More recently, keypoint detectors that utilise the design of the retina to guide new approaches to classification have been developed (Alahi, Ortiz & Vandergheynst 2012, Gomez, Medathati, Kornprobst, Murino & Sona 2015). Key-point detectors also mimic the saccading of the eye to find similar key points. More recently, another approach called the Gabor-Wavelet detector used convolution kernels inspired by biological principles of the visual cortex (Yussof & Hitam 2014).

2.1.3.4 Color-Based Methods

Color within an image gives greater information about the image than do grey values. This has been demonstrated by incorporating colour into Harris corner detectors (Montesinos, Gouet, Cedex & Deriche 1998) (Gabriel, Hayet, Piater & Verly 2005), and other approaches have tried to use a combination of colour via Gradient Vector Flow (GVF) and edge features/contours (Hiremath & Pujari 2008).

Other methods utilise colour-based on existing classification of points by statistical analysis, and show better performance when compared to simpler intensity-based approaches(Stottinger, Hanbury, Sebe & Gevers 2012). Colour has also been used with Spacio Temporal IPs (C-STIPs) to differentiate between features, and has been shown to be at least marginally better than intensity based methods under certain testing conditions (Everts, Van Gemert & Gevers 2014).

2.1.3.5 Model-Based Methods

The model-based approach depends on examples of the features that are to be found. Work done by Guiducci (1988) and, Rosin and Paul (1999) characterised a corner as a blurred wedge, and tried to identify characteristics such as amplitude and angle to match with. These techniques have also more recently been used to help identify more complex features (Moreno, Bernardino & Santos-Victor 2006).

2.1.3.6 Segmentation Based Methods

Segmentation techniques are applied to discover homogeneous regions within an image in order to identify features. This is generally a bottom-up approach aimed at low-level pixel groupings when used to extract features. Segmentation methods are limited because of the large search space of possible feature point groupings. The use of segmentation is complicated by the fact that the most optimal segmentation groupings can be numerous. This makes it very difficult to determine which groupings are the most optimal. Very early applications of segmentation appeared in medical image analysis (Bajcsy 1973, Brice & Fennema 1970). The Canny detector utilised an approach where, after intensity gradients were measured and non maximal suppression was performed, a double thresholding was performed in addition to remove noise from the gradient information (Canny 1986). More recent research demonstrates segmenting of patches into two regions, which is followed by comparing the regions to find corners (Liu & Tsai 1990). The application of this method is severely limited, however. Other approaches were shown to work well with real world images (Malik, Belongie, Shi & Leung 1999).

Most segmentation methods have been unstable and ill-suited to general image processing, though one approach has been shown to surmount these issues (Matas et al. 2004). *Maximally Stable Extremal Regions* (MSER) are extracted. This technique is able to successfully extract homogeneous regions based on intensity and demonstrates good stability. Another variant of this method is able to overcome blurring of region boundaries (Perdoch, Matas & Obdrzalek 2007). A more popular segmentation-based detector is Features from Accelerated Test Segment (Fast), which utilises a number of approaches to form an optimised, fast, yet still generalisable corner detector (Rosten & Drummond 2006), (Rosten, Porter & Drummond 2010).

2.1.3.7 Machine Learning Methods

One of the more popular and well-known machine learning algorithms is based on work by Viola and Jones (2001*b*, 2001*a*, 2004). The algorithm uses a series of Haar-like features that have been specially selected during training with the classifier AdaBoost (Freund & Schapire 1997) to discriminate specially chosen non-deformable objects, such as faces, with a very high success rate. The argument can be made that these types of detectors are not strictly interest-point-based, but the use of discrete Haar features, which constitutes part of a larger feature set, does in itself make the argument that these are points of interest. The drawback of this approach is that even though true positive rates are nearly one hundred percent, there is a low but discernible false positive rate (Sebe, Cohen, Garg & Huang 2005). In addition to this, the use of Haar features means that the process is not rotation-invariant or affine-invariant. Scale invariance is overcome by sampling the area to be processed for faces at a variety of window sizes. Even though this approach is not particularly expensive, it is still a brute force approach. The algorithm also requires very large training sets of positive images (the object to be detected), and negative sets (the object the algorithm is not meant to detect), in the order of tens of thousands, in order to achieve reliable detection rates. Others have expanded on this work to improve the algorithm by improving window processing (Chen, Huang & Fu 2008), and identifying faces by including local context (Schiele & Kruppa 2003), but currently it is still a very specialised (though effective) detection algorithm.

Agarwal et al. (2004) also used a machine learning approach, though didn't achieve the same levels of success as the Viola Jones' work. Their methodology was focused on optimised classification of features via interest points, and adopted a testing and evaluation methodology that utilised true positives and false positives, in conjunction with an F-measure (Rijsbergen 1979), to measure the relative performance of interest points. The selection of features based on information content and cluster comparison was ultimately very time-consuming and difficult to automate.

Other more modern approaches like LIFT (Yi et al. 2016) use marked-up datasets of 3D point clouds with true positives, negatives, and noise, which are then each trained on a Convolutional Neural Network (CCN) pipeline consisting of a detector, orientation estimator, and descriptor, with the ability to perform back propagation. The detector is then trained to create more accurate classification.

2.2 Interest Point Detector Applications

The main purpose of an interest point detector is to pre-process an image to find areas that deserve further processing. Depending on the application, this can be to locate particular features like corners and edges, or it can be more generalised, aiming to find some complex but unknown feature of an area which requires further processing. An interest point detector as a standalone tool is better suited for very simple features. Often, it is used in conjunction with other feature detection algorithms to serve a more complex and purposeful detection strategy (Gil, Mozos, Ballesta & Reinoso 2010, Gauglitz, Höllerer & Turk 2011). In the field over the last two decades, there has been a migration from one type of detector, to more robust forms. These fall into two categories, those known as single scale, and multiscale detectors. As the term implies, single scale detectors are generally optimal at a fixed scale, and are generally invariant to rotation, noise, translation, and illumination, whereas multiscale implementations attempt to stabilise scaling of a scene to preserve feature detections so feature detection approaches are robust under more rigorous conditions. Repeatability of these scale-invariant features is often a motivating factor for such designs and needs to be robust under a variety of conditions.

2.2.1 Measuring Interest Point Detector Performance

Schmid, Mohr and Bauckhage (2000) analysed a number of feature point detectors based on the following criteria:

- Repeatability: The aim of measuring the interest point detector's repeatability is to reliably measure how robust the detector is when an image is subject to a variety of image deformations. Schmid et al. (2000) measured repeatability as being within a certain threshold of where the feature was in a previous image. Repeatability was measured in the following areas.
  - Image Rotation: The image is rotated.
  - Scale Change: The image is at different scales.
  - Variation of Illumination: The image shown darkens and lightens, with uniform and non-uniform illumination.
  - Viewpoint Change: The image is shown at different angles.
  - Camera Noise: Many images of the same scene are recorded and repeatability is measured.
- Information Content: This measures the distinctiveness or uniqueness of each interest point and is analysed by comparing all interest points. If there are many interest points that match, then it could be said that the information content is low, but if there are a low number of matches, then there is a greater chance that those interest points could be distinguished across multiple images, and have a better chance of being repeatable.

Measurements of these criteria with the aforementioned methods are well established and accepted practices, when measuring the performance of interest point detectors (Tuytelaars & Mikolajczyk 2008) (Alahi et al. 2012) (Lindeberg 2015*b*) (Lin et al. 2017). The following is a breakdown of some of the types of feature detectors that have been most well-known or successful in the computer vision field, and are arranged by their broader categorisation based on the types of applications for which they have been best suited. These broader categorisations are single scale, multiscale, image descriptors, and 3D detectors.

2.2.2 Single Scale Detector Basic Principles

One of the early algorithms that could be classified as a feature detector in the computer vision field is the Moravec detector (Moravec 1977), which serves as an example of a single-scale designed feature detector. However it is mostly overshadowed by another more recent example. One of the most reliable and well-used interest point detectors is the Harris detector (Harris & Stephens 1988),which has been used as a benchmark for other detectors well after its initial publication (Loog & Lauze 2010, Schmid et al. 2000, Montesinos et al. 1998). Mikolajczyk and Schmid were able to design a scale-invariant and affine-invariant interest point detector, which integrated the Harris detector, referred to as the Harris-Laplace for the scale-invariant design (Mikolajczyk & Schmid 2004) and the Harris-Affine detector (Mikolajczyk & Schmid 2002). It has also been used as the basis for other detection methods and research over the years since (Baumberg 2000, Mikolajczyk & Schmid 2001, Schaffalitzky & Zisserman 2002, Mikolajczyk & Schmid 2002, Mikolajczyk & Schmid 2004, Gabriel et al. 2005, Tuytelaars & Mikolajczyk 2008, Azad, Asfour & Dillmann 2009, Artieda et al. 2009, Sipiran & Bustos 2011, Nikolic, Rehder, Burri, Gohl, Leutenegger, Furgale & Siegwart 2014, Zhu, Chen & Guo 2014). Since first being described in 1988, it is still one of the better detectors due to its adaptability, and performance.

Harris is an extremely common detector that uses the previously mentioned window processing method, but applies its own particular steps. It is a single scale detector, which has difficulty maintaining robust points as it is rotated. It involves the generation of Gaussian derivatives of the image being processed, as well as applying a Gaussian smoothing and scaling to the pixel values. Equation (2.1) shows the algorithmic representation of the Harris filter (Harris & Stephens 1988). The steps used can be observed below.

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2 \tag{2.1}$$

1. Generate the x and y derivatives of the image.

$$I_x = G_\sigma^x * I \quad \text{and} \quad I_y = G_\sigma^y * I \tag{2.2}$$

2. Find the square of the derivatives at each of the pixels in the image.

$$I_{x^2} = I_x \cdot I_x \quad I_{y^2} = I_y \cdot I_y \quad I_{xy} = I_x \cdot I_y \tag{2.3}$$

3. Determine the sums of the products of these derivatives for each pixel.

$$L_{x^2} = G_{\sigma^2} * I_{x^2} \quad L_{y^2} = G_{\sigma^2} * I_{y^2} \quad L_{xy} = G_{\sigma^2} * I_{xy} \tag{2.4}$$

4. Put at each pixel (x,y) the matrix,

$$A(x,y) = \begin{bmatrix} L_{x2}(x,y) & L_{xy}(x,y) \\ L_{xy}(x,y) & L_{y2}(x,y) \end{bmatrix} \tag{2.5}$$

5. Apply the operator at each pixel, where k is a static value, commonly 0.25.

$$K = Det(A) - k(Trace(A))^2 \tag{2.6}$$

6. K is then thresholded via non-maximal suppression.

Earlier detectors that use A also include Forstner (1986), which was described as,

$$K_{Foerstner}(x) = \frac{Det(A)}{Trace(A)} \tag{2.7}$$

Another detector that shared very similar processing structures to Harris is the Beaudet operator (1978), which uses the determinant of a Hessian matrix to maximise cornerness,

$$K_{Beaudet}(p) = I_{xx}(p) \cdot I_{yy}(p) - I_{xy}^2(p) \tag{2.8}$$

where $I_u(p)$ is the image derivative in the direction of u.

With that, it can be simplified to,

$$H(x,y) = \begin{bmatrix} I_{xx}(x,y) & I_{xy}(x,y) \\ I_{xy}(x,y) & I_{yy}(x,y) \end{bmatrix} \tag{2.9}$$

and be described as,

$$K_{Beaudet}(p) = Det(H) \tag{2.10}$$

Many early examples in edge detection takes on a very similar approach with only slight differences in application. The Sobel (Sobel & Feldman 1968) and Prewitt (Prewitt 1970) operators slightly modify the derivatives that are calculated to enhance edges in the x and y axis. When the derivatives are calculated, like in the Harris detector at equation 2.2, a different kernel is used. For a 3x3 kernel, I_x and I_y become,

$$I_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * I \text{ and } I_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I \tag{2.11}$$

for Sobel and,

$$I_x = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} * I \text{ and } I_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * I \tag{2.12}$$

for Prewitt.

These can then be combined to create a response image using:

$$A = \sqrt{{I_x}^2 + {I_y}^2} \tag{2.13}$$

Thresholded suppression of weaker responses is then applied. The concept of non-maximal suppression was also originally developed by Sobel. As a result, Sobel also has better noise suppression compared to the Prewitt operator.

A much earlier predecessor to Sobel and Prewitt is the Cross edge detector (1963), which also employs a convolution matrix, but which, apart from the difference in the kernel, is very similar to Sobel. The 2x2 kernel,

$$I_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} * I \text{ and } I_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} * I \tag{2.14}$$

helps to discriminate towards the diagonals, as opposed to Sobel and Prewitt's kernels whose focus is on the orthogonal directions. Unfortunately this also makes Cross more sensitive to noise.

The Canny detector (1986) also uses a Gaussian filter to smooth out noise and avoid intensity changes, as well as a more complex double thresholding method and hysteresis to more effectively track the strongest edges.

Canny follows a similar multistep process: firstly, the edges are smoothed by applying a Gaussian filter to the entire image,

$$S(x,y) = G_{\sigma^2}(x,y) * I(x,y) \tag{2.15}$$

using the kernel,

$$S_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} * I \text{ and } S_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} * I \tag{2.16}$$

the gradients of the image are then computed, first by finding the partial derivatives $G_x(x,y)$ and $G_y(x,y)$,

$$\begin{aligned} G_x(x,y) &\approx [S(x,y+1) - S(x,y) + S(x+1,y+1) - S(x+1,y)]/2 \\ G_y(x,y) &\approx [S(x,y) - S(x+1,y) + S(x,y+1) - S(x+1,y+1)]/2 \end{aligned} \tag{2.17}$$

which then allows the magnitude to be computed,

$$G(x,y) = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \tag{2.18}$$

as well as the orientation of the gradient, which is necessary for non-maximal suppression.

$$\theta(x,y) = tan^{-1}\left(\frac{G_x^2(x,y)}{G_y^2(x,y)}\right) \tag{2.19}$$

The Canny edge detector goes beyond simply applying a matrix filter, and also takes advantage of non-maximal suppression by applying a two-stage suppression process, also known as double thresholding, where both the gradient and the orientation of the gradient determine whether the pixel is suppressed. This is used in conjunction with the linking of edges via hysteresis. Non-maximal suppression is done over a 3x3 region, which defines 4 discrete directions d_i, which is then compared to the closest corresponding orientations $\theta(x,y)$. If $G(x,y)$ is weaker than at least two other neighbours along d_i, the position $I(x,y)$ is suppressed

$(I(x,y) = 0)$, otherwise $I(x,y) = G(x,y)$.

Once the image is suppressed, low and high thresholds are used to determine whether the image has false edges that should be further suppressed, and by doing so, also helps to strengthen edges that are relatively weak. Canny is a relatively sophisticated filter that operates on more than one piece of information on the image, such as image intensity, as well as the orientation of edges before it determines suppression of pixels. Furthermore, it uses an additional step to strengthen and weaken the edge being detected based on its own predetermined heuristics. Newer detectors like SUSAN (Smith & Brady 1997) use a circular mask over each pixel, with the center pixel being the nucleus, and the pixels under the mask being grouped according to their similarity in brightness intensity to the nucleus pixel. Corners are determined when the number of pixels, referred to as "USAN" reaches a local minimum and exceeds a specified threshold T. The function $C(r, r_0)$ is used to compare between pixels in the mask and the nucleus,

$$C(r, r_0) = \begin{cases} 1, if|I(r) - I(r_0)| \leq T, \\ 0, otherwise \end{cases} \tag{2.20}$$

with the USAN size being represented as,

$$n(r_0) = \sum_{r \in c(r_0)} C(r, r_0) \tag{2.21}$$

where r_0 and r are the nucleus coordinate and the neighbouring coordinates respectively. This method of corner detection is more susceptible to luminance fluctuation and noise, however.

The Fast detector (Rosten & Drummond 2006) determines corners by applying a segment test to each pixel based on an intenstity candidate pixel I_p and threshold t. If n contiguous pixels in a Bresenham circle with radius r are consistently brighter than the intensity of the candidate pixel $I_p + t$, or all darker than the intensity pixel $I_p - t$, then p is classified a corner. The speed of this test was improved by only testing certain candidate pixels 1,5,9, 13 of the Bresenham circle if $r = 4$. If three of these candidate pixels are brighter or darker than the intensity of the candidate pixel, a corner is found. A machine learning approach is also used based on a decision tree (ID3) to make matching faster and more robust. Additionally, non-maximum suppression is also used to reduce the number of responses. This results in the cornerness measure,

$$C(x,y) = max(\sum_{j \in S_{bright}} |I_{p \to j} - I_p| - t, \sum_{j \in S_{dark}} |I_p - I_{p \to j}| + t) \tag{2.22}$$

where $I_{p \to j}$ represents the pixels positioned on the Bresenham circle. This two-stage design keeps processing fast as the second test is only required on a much smaller subset of points that passed the first test. Given n and appropriate threshold t, the segment test of 16 pixels (where $r = 4$) can be tested (or more optimally tested via an ID3 decision tree), with non-maximum suppression applied to the absolute difference between the pixels in the contiguous arc and n.

2.2.3 Multi Scale Detector Basic Principles

2.2.3.1 Laplacian of Gaussians (LoG)

Laplacian of Gaussians, though not strictly an interest point detector, can differentiate discontinuities in a scene to find regions, or “blobs” of interest. It takes an image I and processes it in a similar manner to Harris in equation 2.2, but with σ as a user defined variable.

$$L_x = G_\sigma^x * I \quad \text{and} \quad L_y = G_\sigma^y * I \tag{2.23}$$

Next, the second-order derivatives are computed,

$$L_{xx} = L_x * L_x \quad \text{and} \quad L_{yy} = L_y * L_y \tag{2.24}$$

For the Laplacian operator, it is defined as,

$$\Delta^2 L = L_{xx} + L_{yy} \tag{2.25}$$

This gives positive responses for dark blobs and negative responses for light blobs where the size is $\sqrt{2\sigma}$, but Equation 2.25 is quite sensitive to the sizes of the features, and the Gaussian smoothing kernel. For multiple scales, this is normalised according to the scale space extrema.

$$\Delta^2_{norm} L = \sigma^2 (L_{xx} + L_{yy}) \tag{2.26}$$

Because of its circular processing, Equation 2.26 is invariant to rotation, and enables for searching of local extrema in image regions more easily, which has been further explored by Lindeberg (2013).

2.2.3.2 Difference of Gaussians (DoG)

The Difference of Gaussians uses a more similar, simplified and less computationally expensive approach. To compute $D(x, y, \sigma)$, it takes the Gaussian $G(x, y, \sigma)$ of two different scales that use a multiplicative factor k of an image $I(x, y)$, where typically $k = \sqrt{2}$ and $\sigma = 1.6$. This form of processing can also be performed without convolution by subtracting adjacent scales as a Gaussian scale pyramid (Burt & Adelson 1987).

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, k\sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{2.27}$$

This allows the searching of what is known as the 3D scale space (Witkin 1987) to find stable features across multiple scales, though both this approach and LoG are sensitive to straight edges, which can create noise. DoG uses what is referred to as octaves (with each octave representing a doubling of σ), which use neighbouring scales within each octave to find the minima, and maxima of the scale space based on 3x3x3 areas around potential keypoints. Generally (Lowe 2004) found that 4 octaves with 5 scale levels each were good defaults.

2.2.4 Image Descriptors

Image descriptors use keypoints to establish a region of interest, so that a feature, or set of features, can be extracted. This involves constructing a classifier that can identify a set of keypoints, then identifying those features in an image, and lastly, extracting characteristics of those features. This could include pixel intensities, contrast, a histogram, or other features like homogeneity or correlation. This process creates what is known collectively as an image descriptor (Lowe 2004, Grauman & Leibe 2011). The field of image descriptors has flourished in the last few decades. Commonly-used image descriptors include, but are not limited to, local binary patterns (LBP) (Ojala, Pietikainen & Harwood 1994), scale-invariant feature transform (SIFT) (Lowe 1999, Lowe 2004), principal component analysis (PCA-SIFT) (Ke & Sukthankar 2004), speeded up robust features (SURF) (Herbert Bay 2006), Weber local descriptor (WLD) (Chen, Shan, He, Zhao, Pietikainen, Chen & Gao 2010), the KAZE feature detector (Alcantarilla, Bartoli & Davison 2012), and the fast retina keypoint detector (FREAK) (Alahi et al. 2012). The attributes of these image descriptors can vary greatly, from invariance to scale, illumination, rotation, and affine transformation. Though this dissertation does not go in-depth on the topic of image descriptors, as it is somewhat beyond the scope of this dissertations's main research goals, the topic still deserves consideration as a matter of completeness with regards to the advances in the field of CV, but also because these detectors still generate keypoints, and thus can function identically to interest point detectors. As such, two of the most well-known, and used image descriptors are covered here.

2.2.4.1 Scale Invariant Feature Transform (SIFT)

SIFT builds on the scale-invariant features developed by the DoG operator (Lowe 2004) covered previously in Section 2.2.3.2, which selects local extrema of the features found by DoG, but then, in addition, further filters the number of candidate keypoints by eliminating points with low contrast, discarding edge responses. This technique borrows inspiration from the Harris algorithm to help determine whether a keypoint is an edge.

For handling low contrast points, a Taylor expansion of the DoG is used to assess each candidate point:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{2.28}$$

where D are the DoG derivatives, and the point being assessed, $\mathbf{x} = (x, y\sigma)^T$, represents the offset from the point. If an extremum's location, denoted $\hat{\mathbf{x}}$, is found by taking the derivative, with respect to $\mathbf{x}$ and making it zero, which is represented by equation 2.29, $\mathbf{x}$ is larger than 0.5 in a dimension. This indicates that the extremum is closer to another sample point. This means the sample point is changed and the interpolation repeated.

$$\hat{\mathbf{x}} = \frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}} \tag{2.29}$$

Based on $D(x)$ representing a pixel range of [0,1], any values where $|D(\hat{x})| < t$ are discarded, with t generally being 0.03 as a default threshold.

For discarding edge responses, a principal curvature is computed with a Hessian matrix of the second-order derivatives of the DoG, as shown in equation 2.30.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.30}$$

This then allows the edges within the image to be properly identified by the ratio of the Trace of the Hessian $Tr(H)$ and the Determinant, $Det(H)$, as shown in equation 2.31.

$$\begin{aligned} Tr(H) &= D_{xx} + D_{yy} = \lambda_1 + \lambda_2 \\ Det(H) &= D_{xx}D_{yy} - (D_{xy})^2 = \lambda_1\lambda_2 \end{aligned} \tag{2.31}$$

The ratio r of the eigenvalues can then be determined (shown in equation 2.32) for each keypoint, with a default threshold of $r > 10$ for determining whether a keypoint is classified as an edge, and thus discarded.

$$r = \frac{\lambda_1}{\lambda_2} = \frac{Trace(H)}{Det(H)}^2 = \frac{(r+1)^2}{r} \tag{2.32}$$

The orientation of the remaining keypoints can then be derived based on the neighbourhood, and over different scales. A 16x16 region around a smoothed version of the keypoint $L(x, y)$ then calculates the magnitudes $m(x, y)$ and orientations $\theta(x, y)$ of each pixel as shown in equation 2.33. Next, a histogram consisting of 8 bins of each of the 4x4 subregions within the 16x16 region is built using the magnitudes and orientations of the 16x16 region around the keypoint.

$$\begin{aligned} m(x,y) &= \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \\ \theta(x,y) &= \tan^{-1}((L(x,y) - L(x,y-1))/(L(x+1,y) - L(x,-1,y))) \end{aligned} \tag{2.33}$$

Each sample point's gradient is weighted according to its distance from the keypoint. Thus, points closer to the keypoint have greater effect on these metrics. This results in a feature vector consisting of 4x4x8=128 elements for a single keypoint that can then determine the dominant orientation of the feature where the keypoint resides. Lastly, the feature vector is normalised to make it resistant to changes in illumination. However, other illumination changes such as saturation or other effects that impact on the gradients are not well handled by normalisation alone, so a threshold can be applied and the gradients re-normalised to handle such effects.

The default SIFT image descriptor attempts to address multiple problems with feature detection by creating a set of approaches to address scale invariance, lighting, and misleading features like edges, as well as to create keypoints with additional information like gradient orientation, which has served well as a multipurpose feature detector (though this has come at the cost of complexity and slow computation time). As a result of this complexity and increased computational cost, alternatives like PCA-SIFT (Ke & Sukthankar 2004) and SURF (Herbert Bay 2006) have attempted to address some of these issues.

2.2.4.2 Speeded up Robust Feature Descriptor (SURF)

The intent of SURF is to act as a faster, more efficient alternative to SIFT. Instead of using Gaussian derivatives, it uses much simpler 2D box filters. It similarly uses a scale-invariant blob-type detector, but is based on the determinant of the Hessian matrix to deal with scale selection and to find keypoints. It approximates second-order Gaussian derivatives efficiently by using integral images and box filters, similar to the Haar features used by the Viola-Jones detector (Viola & Jones 2004). These approximations are represented as D_{xx} D_{yy} and D_{xy}, with the determinant of Hessian represented as,

$$det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{2.34}$$

.

w represents the relative weight that is applied to the filter response. This results in the blob response of the image. Like SIFT, these approximations are also done at various scales, with octaves, but by applying box filters of different sizes, a 9x9 box becomes the general representation of a Gaussian filter at $\sigma = 1.2$, and local maxima are found using quadratic interpolation. For each scaled image, larger and larger masks are used to approximate what the Gaussian filter output would be. Non-maximum suppression is applied to each possible keypoint in a 3x3x3 area across scales to find stable keypoints, which works to emulate the processing that SIFT's use of LoG utilises.

SURF then creates a square 20x20 processing window around the candidate interest point, which is divided into 4x4 sub regions. Each region has Haar features (d_x, d_y) applied to it in the vertical and horizontal axis respectively to compute 5x5 regions of points. The responses from d_x and d_y for each 4x4 sub-region are then summed to represent the vector v,

$$v = (\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|) \tag{2.35}$$

This creates two points that can be plotted in 2D space, and acts as the orientation for each respective 5x5 region. Like SIFT, they are weighted according to their distance from the central keypoint.

With this done for each 4x4 region, the descriptor represents 4x4x4=64 elements for each keypoint. The descriptor is also normalised to minimise the effects of illumination. This type of processing results in a much faster image descriptor than SIFT, with the tradeoff that it is not as reliable, or accurate. Other limitations of SURF include less robust features against rotation in both 2D and 3D situations, especially when the rotation is extreme, and less affine invariant compared to SIFT (Pang, Li, Yuan & Pan 2012)

2.2.5 3D Interest Point Applications

There have also been attempts to utilise 2D detectors to reconstruct real scenes based on calibrated photos (Mozos et al. 2007, Labatut, Pons & Keriven 2007, Artieda et al. 2009) but such attempts have been heavily dependent on properly calibrated camera positioning to generate reliable point clouds. In other cases, points of interest are filtered based on a brute-force approach where very large, precisely calibrated datasets are used to find intersecting pixels that share common intensity, or colour-based information (Agarwal, Snavely, Simon, Seitz & Szeliski 2009). Moving beyond algorithms based solely on 2D imaging to generate 3D interest points, attempts to leverage the advantages of 3D spaces, where ground truth is well established (especially via 3D scanned models), are also being carried out (Gupta, Gupta, Singh & Wytock 2008, Rahmani, Mahmood, Huynh & Mian 2014). A survey by Hänsch et al. (2014) outlines and compares a number of recent forays into leveraging the ground truth of a virtual space using 3D models, 3D keypoints, and point cloud fusion via Normal Aligned Radial Feature (NARF) (Steder, Rusu, Konolige & Burgard 2011) and a 3D variation of SIFT (Flint, Dick & Van Den Hengel 2007). Though there is a sincere attempt with these algorithms to use 3D information of a scene's depth (Zhu et al. 2014), the approaches to repeatability evaluation closely mirror that of Schmid et al.'s earlier work (mentioned in 2.9.2) in relation to ratios of true positives and true negative sets of points, and are heavily dependent on the scene's ground truth (these detectors require knowledge of scene depth to operate) to classify features within the scene. A 2D implementation cannot be used, which effectively renders such 3D detectors inoperable in 2D contexts.

A more recent survey of 3D detectors by Guo et al. (2016*a*) also highlights that many 2D feature detectors fail to utilise depth in order to analyse the performance of 2D points in 3D environments. In order to reconcile the evaluation of 3D detectors that did not share common evaluation criteria, Guo adapts the recall and (1-precision) by Mikolajczyk and Schmid (2005) to better suit the comparison of 3D point performance (which utilised surface histograms to describe the point's features) under a variety of scene transforms. A variety of 3D models from 8 different datasets were tested on 10 3D based feature detectors, SI (Johnson & Hebert 1999), 3DSC (Frome, Huber, Kolluri, Bülow & Malik 2004), LSP (Chen & Bhanu 2007), THRIFT (Flint, Dick & Van den Hengel 2008), PFH (Rusu, Blodow, Marton & Beetz 2008), FPFH (Rusu, Blodow & Beetz 2009) SHOT (Tombari, Salti & Di Stefano 2010*b*), USC (Tombari, Salti & Di Stefano 2010*a*), RoPS (Guo, Sohel, Bennamoun, Lu & Wan 2013*b*), TriSI (Guo, Sohel, Bennamoun, Lu & Wan 2013*a*). By copying Mikolajczyk and Schmid's evaluation approach and adapting it for 3D detectors, Guo was able to empirically differentiate the performance of each 3D-based feature detector. However, the relative performance of 2D feature detectors were not measured or compared. Additionally, based on the testing methodology, there is a strong coupling with the ground truth of the scene and the detector, such that 3D-specific information (such as the normals of 3D models) is required in order to properly classify features for detection.

In many cases, there is a strong delineation between 3D detectors that depend on a 3D ground truth to establish performance, and 2D detectors that require extensive marking up of scenes to establish a ground truth or a greatly simplified abstraction of the scene via a homography, which estimates the 3D pane of points relative to another perspective so that they can be transformed. With the exception of LIFT (Yi et al. 2016), there are few examples of hybrid detectors that are optimised using 3D data but operated in 2D contexts, and thus do not require a 3D ground truth. However, even LIFT is trained using predetermined, marked-up 3D features datasets based on a real world environment, and measures performance using a ratio of true positives and true negatives when evaluating its performance compared to existing detectors, consistent with the conventional performance evaluations described by Schmid. That is to say, it trains itself on predetermined true positives and true negatives only.

2.2.6 Commonalities in 2D Interest Point Detectors

Most of the keypoint detectors described to this point share a fairly common layout. They are more commonly referred to as single scale detectors. A common theme among them is that they use the intensity values of an image, along with a processing window, to process areas in an image, to expose features such as corners or contours. Generally, anything that uses a kernel or mask would be using a type of scanning window. A classic example of these detectors is the Harris detector (Harris & Stephens 1988) which, even in recent times is still used as the basis for other areas of research (Mikolajczyk & Schmid 2001, Schaffalitzky & Zisserman 2002, Mikolajczyk & Schmid 2002, Mikolajczyk & Schmid 2004, Sipiran & Bustos 2011), and as the basis of other more advanced scene processing (Nikolic et al. 2014, Zhu et al. 2014, Dawn & Shaikh 2016). In simplified terms, these types of detectors use an image I, and a scanning window W of size $n \times n$. For every pixel p in image I that the scanning window W is focused on, apply a filter K to each p in the image I to create a response output. Once the interest image I^* of these responses is created, all but the strongest responses are suppressed via non-maximal suppression, so that only the strongest responses in the scanning window are used. These would be considered interest point candidates, and may then be further filtered for validity by being measured against an empirically-determined threshold h. So the steps are:

1. Scan the the image using a scanning, or sampling window. As the scanning window W moves across the image, process the image data according to the predetermined filter being used.

2. Take the data resulting from this processing and create a “response image” that shows the newly processed pixel “responses” based on the filter.

3. Use a thresholding function that arbitrarily suppresses all but the strongest responses over a localised area, similar in nature to the processing done via the scanning window.

The strongest responses become the “points of interest” that constitute interest points for that particular image.

$$K(p) > sup\{K(p_w)|\forall p_w \in W, p_w \neq p\} \wedge K(p) > h \tag{2.36}$$

This process, shown in equation (2.36), outlines the basic structure of most traditional detectors today, and is very widely used. Most commonly it is the pixel value that is intensified or suppressed in order to find features, and algorithms that follow this general structure could be classified as interest point detectors. Even more sophisticated image descriptors like that which SIFT and SURF creates share similarities, although they are much more complex compared to older varieties.

These fundamental similarities make it possible to structure algorithms such that they can be represented as a set of steps that sequentially process the image to create a response image, which can then be further filtered based on thresholding, or another metric, and the resulting points used as candidate interest points for further processing. This will be covered in more detail in Section 2.7.

2.3 Overview of Evolution in Computer Science

Dawkins (1983) asserts the theory of universal Darwinism, which is a more generalised extension of biological evolution, claims that the observation made by Charles Darwin in nature is the way in which all life comes into being, and that evolution exists in all forms of nature. Plotkin (1993) further claims that universal Darwinism can exist outside of nature, not only as a purely physical phenomenon, but as the basis for all intelligence, the notions of science, the workings of our brains, or more abstract notions such as machine learning. Plotkin reduces this to a three phase process of generation, testing and regeneration, or g-t-r. What follows in this section is an overview of how learning strategies observed in the natural world have been applied in the field of computer science. Though by no means a complete review, it is a background of the progress in the field.

2.3.1 Evolutionary Computation

Evolutionary computation (EC) is a process inspired by biology, and has been successfully applied as a learning or optimization tool for computers (Bäck, Rudolph & Schwefel 1993, Thomas Bäck 1993, Bäck 1996, Bäck, Fogel & Michalewicz 1997). It has been used as a tool in computer science for over 50 years (Mitchell 1996). It is based on the abstract concepts of Darwinian evolution, and other evolutionary models, and has been applied to successfully solve complex AI problems. The solutions created are often complex, non-intuitive in their design, and beyond the scope of human designers. These nature-inspired processes have been re-designed and improved so that EC-based algorithms, more commonly referred to as evolutionary algorithms (EAs), can solve problems that are far beyond the understanding of human designers. EAs provide, in many cases, solutions that can solve seemingly impossible problems, its adaptiveness and impressive problem-solving ability has succeeded in providing solutions, where human ingenuity has faltered. Another related branch of this field, known as swarm intelligence (SI) (Eberhart & Kennedy 1995, Bonabeau, Marco, Dorigo, Théraulaz, Theraulaz et al. 1999) also emulates the behaviours found in nature to solve complex problems. Swarm-based approaches most commonly use a population of simple agents to intelligently organise solutions and are modelled on animal behaviour such as ants or bees (Dorigo, Maniezzo & Colorni 1996, Karaboga & Ozturk 2011), or even flocks of birds (Beni & Wang 1993, Bonabeau et al. 1999). There are many different types of EAs, but most share a number of basic principles. Each begins with a population of potential solutions (also informally known as individuals). Small changes are made to the solutions in the population, and prefer changes made to solutions that are considered, based on objective measures, to be more "fit". Charles Darwin was the first to observe this process, stating "This preservation of favourable individual differences and variations, and the destruction of those which are injurious, I have called Natural Selection, or the Survival of the Fittest." (Darwin 1872). Through this process, it is hoped that the potential solutions will slowly "evolve" into better solutions over time. All EAs share the underlying Darwinian principle of "survival of the fittest" in at least some way, and it is this principle that helps drive an EA to solve problems. In the field of Computer Science, this is translated into what is known as the fitness function.

2.3.2 Swarm Intelligence

Swarm intelligence draws its inspirations from the behaviour of living organisms in the environment to organise and search. The "swarm" consist mainly of a population of simple agents that are able to communicate, emulating communication styles such as ants (Dorigo et al. 1996), birds, fish (Eberhart & Kennedy 1995), bees (Karaboga & Akay 2009, Karaboga & Ozturk 2011), and even plants (Baluška, Lev-Yadun & Mancuso 2010). In the field of SI, particle swarm optimisation (PSO) and ant colony optimisation (ACO) are two of the major algorithms that have been extensively utilised.

PSO, devised by Eberhart and Kennedy (Eberhart & Kennedy 1995, Kennedy 1995) sought to emulate the socialisation interactions of animals in the environment. In line with the behaviours of an EA, it uses a population of simple agents, more commonly known as a swarm, or particles, with a simplified set of basic behaviours, so that it can search the problem domain. The swarm/particles are randomly generated, and then evaluated according to a predefined set of criteria. Based on its local environment, the particles adjust themselves accordingly, with an individual and global fitness measure to gauge the optimisation of the swarm at an individual and global level(Poli, Kennedy & Blackwell 2007).

ACO was originally described by Dorigo (Dorigo et al. 1996). It uses a different approach to PSO when it comes to search strategy, which is more optimised for pathfinding (Dorigo, Maniezzo & Colorni 1991). It closely emulates the behaviour of foraging ants as the swarm population, and utilises an artificial pheromone trail which evaporates over time and determines the direction in which the swarm moves. Each ant agent uses stochastic decision making to explore the search space. The evaporating pheromone paths prevent sub-optimal paths from persisting, and strengthens paths where more ants pass, hence resulting in an optimisation in the shortest pathways.

Other SI algorithms, like artificial bee colonies (ABC) (Karaboga & Basturk 2008, Karaboga & Akay 2009), have roles for certain particles in the swarm, such as employers, onlookers, and scouts, with food sources being potential solutions. The employed bees search for food nearby based on their memory, onlooker bees follow employed bees to these food sources and fitness of the food sources is gauged by the onlooker, while scouts are elevated from onlookers once they run out of a food source, and search for new sources.

Additional SI algorithms include the grey wolf optimiser (GWO) (Mirjalili, Mirjalili & Lewis 2014) and bat swarm optimisation (BSO) (Yang 2013).

Other EC algorithms that are worth mentioning, but reach beyond the scope of this dissertation, include differential evolution (DE) (Storn & Price 1997, Price, Storn & Lampinen 2006), learning classifier systems (LCS) (Holland, Booker, Colombetti, Dorigo, Goldberg, Forrest, Riolo, Smith, Lanzi, Stolzmann et al. 1999, Wilson 1999), evolutionary multi-objective algorithms (EMO) (Coello, Lamont, Van Veldhuizen et al. 2007), and artificial immune systems (AIS) (Castro, De Castro & Timmis 2002).

2.3.3 Basic Concepts of an Evolutionary Algorithm

When describing an EA, it can be difficult to abstract away the nuanced differences among all the EAs that have been developed. Most EAs differ in terms of how individuals are represented, and how populations are formed, altered and selected. By generalising these aspects of EAs, we can grasp what all EAs share. A generic description of a very simple EA is shown as pseudo-code in 2.3.3, based on

(Wiegand 2004). Following that, each of the major functions will be described in greater detail. Most EAs use this algorithmic structure as a basis for their own algorithm.

Basic Evolutionary Algorithm

```
1. Initialise population of individuals
2. Evaluate population
3. $t := 0$
4. do
        4.1 Select parents from population
        4.2 Generate offspring from parents
        4.3 Evaluate offspring
        4.4 Select survivors for new population
        4.5 $t := t + 1$
   until terminating criteria are met.
```

Initialisation

The first step in the algorithm is to generate a set of candidate solutions through some heuristic means. Initialisation can be executed in a variety of ways, but usually results from each candidate solution being randomly generated. This in itself does not necessarily create any type of solution (or even a good one), but via evaluation of each individual's fitness, some individuals will be seen to perform better than others (even if this performance is very bad), and a starting point for optimisation will be achieved. Random generation is often preferred because it adds diversity, and because, through random distribution, each individual will be exploring a different part of the solution candidate's possible search space. This means that many individuals will be approaching the problem from a variety of different perspectives, so to speak.

A program's search space is in reference to all possible program designs that a program could have. A search space is often referred to in optimisation and searching algorithms. Often in search algorithms it represents all possible values that must be analysed to find the correct value. In evolutionary learning, it carries a similar meaning, but instead of values being searched, program designs are searched. EAs search through all possible program designs until a correct, or close enough, design is found.

Evaluation

Evaluation is one of the most variable aspects of EC, because the way individuals are evaluated is entirely dependent on the problem that the individuals are being optimised to solve. Every problem requires the design of a different fitness evaluation. The evaluation of individuals must be designed carefully to properly measure each individual's fitness. Compounding this, fitness evaluation is usually the most computationally expensive aspect of the EA. Commonly, it requires the solution to be decoded, or compiled and run, in order for the individual to be properly evaluated. Evaluation, from an EC perspective, almost always refers to measuring how well optimised an individual is, in comparison to the rest of the population. Evaluation usually consists of one or more objective tests on which each individual is tested. These results are then used to quantitatively measure how well each individual performed, compared to the rest of the population.

Selection

Selection involves choosing parents to create new offspring. A bias is made towards individuals with better fitness over those that are less fit. Two common parent selection methods are roulette selection and tournament selection. Roulette selection gives each individual a chance to produce offspring based on their fitness value, with more fit individuals having a greater chance of producing offspring. Tournament selection involves taking a pair of individuals at random and choosing the best of the two based on their fitness value.

Generating Offspring and Evolutionary Operators

The creation of offspring in an EA is similar to biological creation in nature. Creation of offspring uses operators that preserve the heredity of the offspring by taking material from each parent. Thus, each individual is a culmination of the best aspects of each previous generation of individuals. EC refers to the operators that create offspring as "genetic operators". Genetic operators often fall into one of two groups, defined by the following processes: *Recombination* takes parts of two or more parents and creates a new individual. Recombination concentrates on the genotype of an individual, which represents the changes made as a result of the building blocks the individual receives from their parents. *Mutation*, mimics the biological effect the environment has on the individual, also referred to as the individual's phenotype.

Termination

The way in which the EA stops can be set a number of different ways. Most commonly, the EA will stop after a number of iterations (more commonly referred to as generations). Another stop condition is to continue running until an individual or general population reaches a fitness threshold. Decisions regarding which stopping conditions to employ can be dependent on many factors, such as population size, the type of problem being solved, and the nature of the end solution required from the EA.

2.4 Types of Evolutionary Computation Algorithms

This section aims to outline the types of EAs that are used in EC. There are arguably four accepted subtypes of EA that exist in the field. These are:

- Genetic Algorithms
- Evolution Strategies
- Evolutionary Programming
- Genetic Programming.

Though some disagree that these four are the currently correct categorisations of EAs, each of these four has a distinctive approach when using evolution in EC. The main algorithm applied in this dissertation is *Genetic Programming* (GP), which will be described in more detail, whereas the other categories will only be briefly touched on here, so as to give a broader understanding of the field as a whole, and properly clarify where the field of genetic programming stands in relation to the other similar fields of research in EC.

2.4.1 Genetic Algorithms

Genetic algorithms (GAs) are a category of evolutionary algorithms that closely mimic in some way the biological processes that occur in nature. Genetic algorithms embrace the genotype-phenotype characteristics, but avoid more specific biological aspects (Holland 1992). GAs usually generalise biological behaviour, and adapt this to EC environments. Recombination, which represents the exchange of an individual's genotype information to create new offspring, is a well known biological process that has been adopted by numerous GAs and implemented for use in computers.

2.4.2 Evolution Strategies

Evolution strategies (ES) aim to harness the concept of evolution in a much broader sense. They first appeared in the 1960s as strategies for optimising the parameters of aerofoils (Rechenberg 1965). They concentrate on primarily using mutation to improve populations. ES use a much more generalised approach to evolution compared to GAs, which adopt the concepts of an individual in a more biological sense. ES is specifically related to recombination, phenotypes and genotypes.

2.4.3 Evolutionary Programming

Evolutionary programming (EP) is similar in some respects to ES, but still has distinct differences (Bäck et al. 1993). One distinctive characteristic of EP is that it does not use recombination. Additionally, unlike GA, EP does not require a codified representation. When it comes to selection, EP often uses a tournament-based selection process, whereas ES typically selects a cutoff point at which to discard individuals.

2.4.4 Genetic Programming

Genetic programming is a computation process that can create solutions to problems unassisted.

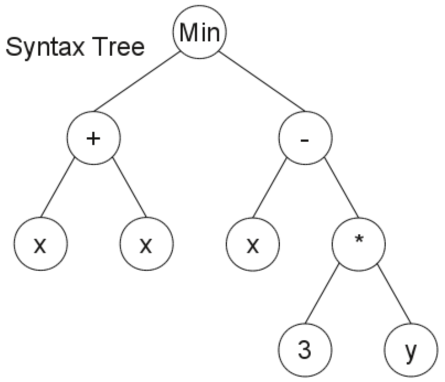
Genetic programming has become an effective machine learning tool, which can produce many viable solutions in different problem areas. Genetic programming is used to evolve a population of computer programs. After a population is created, each individual within the population is tested to determine how effective it is at reaching a pre-defined solution. Each program's result is quantified, usually as a single value that determines the overall fitness of each individual within the population. Once this fitness has been determined, slight variations are made to each individual within the population to produce new and hopefully better offspring. The ways in which these offspring can be created can vary, but usually involve mutations of the individual, or a mix of two parent individuals, to create an entirely new individual, termed a crossover. The process is then repeated until a certain number of generations has transpired, or until one of the individuals within the population meets the performance criteria given.

2.4.4.1 Genetic Programming Representations

When using a Genetic programming EA, the programs are formatted as a tree-like structure. The entire solution is a set of linked n-ary nodes, also known as a syntax tree. Each node consists of two or more branches that link to other nodes. The number of nodes that can be linked to from an existing node can be fixed or arbitrary, based on the contents of the node. Each node that exists in the tree contains a primitive that is either defined as a function or as a terminal, with functions only being able to link to terminals, and terminals only being able to link to functions.

This structure makes it easy to manipulate the solution without breaking existing functionality, and helps to maintain a limited degree of soundness when the solution is compiled and run. A typical tree can be very complex, yet large parts of the program structure can be easily redesigned with minimal concern for the program's stability. Figure 2.1 illustrates how a candidate solution `min((x+x),(x-(3*y))` would be represented as a syntax tree. The implementation of this structure can vary greatly depending on the problem, as well as the need for efficiency or the programming environment. Generally, the number of arguments on each node is fixed to a specific number (Poli, Langdon &

Figure 2.1: Example of a syntax tree representing `min((x+x),(x-(3*y))`.



McPhee 2008).

An alternative to tree-based representations is to use a linear representation (Eklund 2002). Doing so can speed up the algorithm by orders of magnitude, since the algorithm is using lower-level functions compared to higher-level programming languages. A second reason is because computers are not optimally designed to run tree-based programs. Evolving binary code without the assistance of a tree-based format greatly reduces the amount of overhead processing required. It also makes the linear programs easier to process and analyse. One such instance demonstrates that the linear code could be easily scanned and analysed by automated functions for "dead code" segments, to help optimise the programs (Brameier & Banzhaf 2001). Poli (Poli et al. 2008) also cites many sources indicating that linear GP provides better analysis of search spaces, as does a different representation called graph-based GP. Graph-based GP includes variations such as parallel distributed GP (Poli 1996*a*, Poli 1999), parallel algorithm discovery and orchestration (Teller 1996), and Cartesian GP (Miller 1999). The drawback of linear GP implementations is that they are very architecture-specific, but in some cases this can be outweighed by the significantly good performance, and other improvements that linear based GP can provide, as mentioned previously.

2.4.4.2 Function Sets and Terminal Sets

When it comes to defining programs in GP, the language that is used to create these programs must be defined. The language is typically classified into terminal sets and function sets. The primitives that are defined in each of these sets provide the basis for how the computer programs can be created (Poli et al. 2008). Poli also discusses the necessity for closure and sufficiency when choosing primitives for each of these sets that make up the predefined language.

Terminal sets can be variables, functions without arguments, or constants. Variables are usually placeholders for external outputs when the program is run. Functions without arguments usually return a value that will be different each time, such as a function that returns a random number. Constants are predefined or random values that can be created or modified (via mutation) when the program is created.

The function set is largely dependent on the type of problem that needs to be solved. Typical function sets could consist of (+, -, *, /), but in no way is the function set dependent on, or restricted to, these particular functions. The primitives that make up a function set could also be predefined functions that receive one or more inputs. A useful analogy to describe the purpose of the function and terminal set is to imagine the function set as applied to a collection of Lego pieces. The function or terminal set in this instance would define what types of pieces are available to use among the stack of Lego at your disposal.

In order for the function set and terminal set to function together properly in a syntax tree, they must have closure (Koza 1992). Closure requires the program to be type-consistent and evaluation-safe. Type consistency means that all the defined functions should be usable with any of the primitives defined in the terminal set. The reason for this is that crossover will regularly and arbitrarily move one part of a tree to other areas of the program's tree. This means that all primitives in the function set must be able to return values of the same type, eg. +, -, * and /, can all take two integers, and return an integer. Evaluation safety means that when a program is run, it should not be in danger of failing. Examples include instances where the program divides by zero. This usually requires certain checks to be in place to gracefully fail, or handle the situation appropriately so that the program can continue running.

Sufficiency also has an impact on the types of primitives that are selected. Sufficiency means that it is possible to create a solution to the provided problem based on the primitives provided. Determining sufficiency in this case is difficult. It can require in-depth understanding of the problem, or sound theory to know whether the selected primitives are sufficient. However, if the primitive set is not sufficient or sufficiency is unknown, it is still possible to create solutions that are good enough.

Initial Populations

Once the main primitive sets are decided, the GP process is ready to generate initial populations. The initialisation of populations is usually done randomly. The type of randomised population that is generated will influence how effective and efficient the subsequent optimisation can become. It is important that when initial populations are generated, there is enough diversity in the population (Monsieurs & Flerackers 2003). Initialisation can affect the overall size of the solution, bloat (also referred to as uncontrolled growth), diversity, and the time taken to find an acceptable solution (Monsieurs & Flerackers 2003). The type of initialisation method used will likely influence subsequent optimisations. Two of the earlier and better-known methods used to randomly generate a population are grow, and full. Another variation of these two methods is called ramped half-and-half (Koza 1992), which incorporates the grow and full approaches in equal measure. Seeding with existing solutions will also be explored.

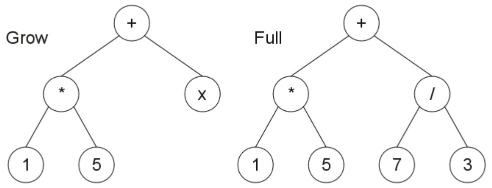
Grow, Full and Ramped Half and Half

Each of these methods generates randomised trees up to a user-specified depth. The depth of a tree is defined as the number of nodes that can be traversed away from the root node, with the root node having a depth of 0. Most often, GP trees are shown with the root node at the top and the nodes moving downwards, like an upside-down tree. The nodes at the very end of the tree are referred to as "leaves" and the size of the tree refers to the total number of nodes (including the root node) that exist in that tree.

The grow method creates trees of various sizes and shapes. Unlike full, nodes are randomly selected from the whole primitive set until the depth limit is reached. At the depth limit, only terminals are selected for the leaves of the tree. Once the depth limit is reached, the random generation is stopped before the rest of the nodes are populated. This results in many of the trees being of different sizes and shapes. Any existing function nodes without terminals are populated as leaves, to ensure evaluation safety.

Full functions similarly to grow, except that all available nodes are filled to the depth limit. This usually results in all trees being "full" to the same depth, but not always. However, if the available functions in the function set differ in the number of their arguments, or have mixed arity, then this will not always be the case.

Figure 2.2: Examples of a randomly generated grow and full tree with a depth of 2.



Each of these methods is not in itself a very good way of randomly initialising a population. Neither grow nor full is good at providing a wide variety of shapes, since grow will mostly produce trees that are predisposed to being small in overall size, as the method usually results in one side of the tree being filled while the other side is not. full, on the other hand, will very often have overly large-sized trees. Ramped half-and-half incorporates both grow and full, to create a more diverse mix of tree types. This method is simple in principle: one side of the root node is generated using the `grow` method, and the other side is generated using the `full` method.

Even when using ramped half-and-half, it can be difficult to control aspects of generation such as tree size and shape. Problems associated with `grow` make it highly sensitive to the function and terminal sets used. If the `grow` method has more terminals than functions, then the depth of the trees will be much smaller, regardless of the depth limit. Likewise, if the number of function primitives outnumbers the number of terminal primitives then `grow` will behave similarly to `full`. The behaviour of `grow` is also different, depending on the number of -arities in the function set. The multi-arity of functions in the set could make tree sizes and depths differ greatly when `grow` is used. Langdon (2000) has also shown that ramped half-and-half does not effectively explore the search spaces of programs that result in long, thin, trees.

The Impact of Population Initialisation

When searching for a possible solution, the search space can be considered infinite. To search for all possible solutions in a uniform manner is going to be impossible. Therefore, any population that is generated will have some form of bias attached to it. Ramped half-and-half creates rather bushy trees which are better suited to symmetric problems, whereas the Santa Fe ant trail's solution is more randomly structured (Langdon & Poli 1998). What this implies is that the nature of the problem involved can direct the method employed to generate the initial population. Another contributing factor that could affect the optimisation process can be the size of the trees themselves. Sometimes a lot of time can be wasted exploring simple solutions to a problem that requires a much more complex structure. Chellapilla (1997) found that better results could be achieved when tree size was better controlled.

Iba (1996) and Bohm (1996) demonstrated that trees can be more precisely controlled. However, these alternatives were computationally complex (Luke 2000). Langdon (1998) developed the ramped uniform initialisation method, which creates uniform trees at user-specified ranges. The generated trees are asymmetric and more variable, compared to those that were generated with ramped half-and-half. Trees generated using this approach can have leaves very close to the root node. This can benefit solutions where particular variables play a more dominant role, and helps ensure the search space is more effectively explored at initialisation.

Bloat can also be caused during the initialisation phase. Crossover can prematurely cause uncontrolled growth in programs. The cause of this can be present before the first generation has even started. Initialisation methods that produce many short programs can result in uncontrolled growth, a number of generations later (Dignum & Poli 2007). This is because crossover performs better in a more diverse, non-uniform tree size distribution (Poli, Langdon & Dignum 2007). Uncontrolled growth can be avoided by analysing the initialisation before any runs are started. This is known as Lagrange initialisation. It involves performing many rounds of crossover on the population and analysing the population for signs of bloat.

Other methods of initialisation that don't use randomisation are called seeding. This approach seeds the population with an individual. This individual, though not a solution in itself, is considered a good starting point. The seed may have been a previous solution, or a human-designed solution. In the case of human-designed solutions, Marek, Smart and Martin (2002) found that designed programs were not robust when competing against many other individuals within the population. They speculated that this was because the designed seed was too far ahead performance-wise, yet they feared that a seed that was on par with the performance of the rest of the population would mitigate any gains the seed originally had.

Marek, Smart and Martin (2002) highlight the negative aspects of using a seed in a population. Often, the seed will outperform many individuals, and come to dominate the population as its descendants are preferred over other promising but less optimal individuals. This can result in early convergence, while population diversity drops rapidly. Conversely, the seed may find itself quickly removed from the population because of its lack of robustness in the environment. One way of countering these problems is to use the seed as a template for all the other individuals in the population. The population could consist of identical individuals, or all the individuals could be mutated copies of the seed. Although the population's diversity is greatly reduced through this process, and the program's possible search space is greatly curtailed, this will help these individuals to create children that will preserve traits that aid in performance in subsequent generations.

Defining the Fitness Function

The fitness function is one of the most variable aspects in EC. How it measures fitness is largely dependant on the problem it is meant to solve. The fitness function's main purpose is to evaluate each individual within a population by subjecting the individual to one or more objective tests, then rating how well the individual completes these tests. It is very important that the fitness function test each individual's performance thoroughly to avoid situations where the solution is overtrained to only meet certain criteria that are tested during evolution, while failing in other areas when used in real-world scenarios. Such a lack of robustness would usually result from the fitness function's failure to sufficiently test each individual properly on every aspect of the problem the individual is meant to solve. This is usually because the problem is not properly analysed, and because appropriate objective tests are not designed to measure performance accordingly.

The fitness function is also one of the most computationally complex aspects of EC since, in many cases, each individual must be compiled, run and tested thoroughly. Designing an efficient evaluation procedure is a major consideration when designing the fitness function. The other consideration is that often the fitness function cannot objectively measure each individual's performance. This is because there are simply too many, or perhaps an infinite, number of situations to which the individual could be exposed in which it must be able to output the appropriate response.

Multi-objective genetic programming (MO GP) aims to provide methods for designing fitness functions that take multiple fitness criteria and determines the best individual according to all the fitness criteria simultaneously.

Selection of the Fittest

When it comes to selection of individuals, tournament selection is seen to be the most popular selection strategy in GP (Poli et al. 2008).

Tournament selection selects a number of individuals at random from the existing population. The individuals are compared according to their fitness, and the best is selected to persist in future generations. This type of selection is very different to selecting the top tier individuals according solely to the fitness measures. Tournament selection's frame of reference is restricted to the selected individuals. The rest of the population is ignored when it comes to validating whether the selected individuals are more or less fit.

This process reduces selection pressure among individuals in the population. Selection pressure that is strong in a population is highly biased towards the population's most fit individuals, whereas weak selection pressure discriminates less between the fitness levels of each individual. By making selection weaker, the selection process is less biased, and helps to prevent one extremely good individual from dominating the population with its exclusive offspring. In contrast, with high selection pressure, this quickly eradicates diversity and potentially stunts any further optimisation. Tournament selection also lends bias towards slight improvements, so that even relatively weaker individuals' improvements will persist in future generations.

GP Crossover and Mutation

In GP, once the individuals have been analysed, the best individuals are taken, and changes are made to their structure in order to improve their performance further. Crossover and mutation are the two major methods in which a program can be modified. Each method can approach crossover and mutation in a myriad of different ways.

GP Crossover

Crossover operates by choosing two fit individuals in the population and creating an offspring individual that is made solely from the two chosen parent individuals' structures. This is also referred to as recombination.

The most common form of crossover is called subtree crossover. A node in each of the parents' program trees is selected, and named the "crossover point". Then, each parent's selected subtree is switched at the crossover point, to create new individuals. Crossover has been proven to be very effective at improving the fitness of individuals in a population (Koza 1992), but has a number of drawbacks. One drawback is that most trees have a 2 to 1 ratio of leaves to nodes. Thus random selection of crossover points can lead to most nodes only containing leaves, resulting in very little information being exchanged. Koza (1992) suggests predisposing selection of functions in 90% of cases and leaves in 10% of cases, to avoid simply swapping two leaves in the tree.

One-point crossover (Poli & Langdon 1998*b*) works to preserve the positions where the trees exist in the parent individuals, thereby preserving some of the existing functionality of the parent trees. This method analyses each parent, and finds tree positions that exist in both parents. One of these is chosen as the crossover point; everything above this crossover point uses parent 1's tree and everything below the crossover point is replaced with parent 2's tree.

Uniform crossover (Poli & Langdon 1998*a*) is another version of crossover, similar to one-point crossover, in that it finds the common tree structure shared by both parents. This method, however, traverses each commonly shared node, and randomly chooses whether the new offspring's parents' commonly shared node should inherit from the first or the second parent. This helps to avoid bias when the algorithm is traversing the search space, which can happen when the majority of nodes in a tree are leaves, and also avoids local search bias (i.e., the situation in which, the program's entire search space is not being properly explored). A failure to optimise can happen when the crossover process changes little in the new offspring tree's structure as a result of concentrating on one single crossover point to create a new child, instead of many crossover points.

Context-preserving crossover (D'backhaeseleer 1994) is employed to identify a number of points in the tree that exist in both parents, much like one-point crossover. Then, only the commonly identified points can be used to identify possible subtrees that can be used for crossover. This has the added effect that movement of certain parts of a subtree is restricted in terms of where they can be placed in the new offspring, especially in moving subtrees to different levels in the new offspring tree. This limits diversity a great deal, but D'Haeseleer (1994) found that when combined with regular crossover, performed better than regular crossover.

Size-fair crossover (Langdon 1999) randomly selects the node that is to be the crossover point, and calculates the size of the subtree in the first parent. Then, in the second parent, a subtree of the same size is found. This helps to preserve the structure of the tree to a limited degree, but mainly helps to preserve the overall size of the offspring. Langdon (1999) found that the size of the programs could be significantly reduced while still maintaining performance of the solutions produced. The number of levels in these trees could increase no more than one level each generation. As a result, the bloat of theses trees was reduced significantly.

GP Mutation

Mutation is used in GP to improve the performance of an individual in the population by making small changes to parts of the tree. The use of mutation also acts as a way of introducing diversity into the population (Koza 1992). This can also be achieved via crossover, but mutation affords the opportunity to introduce variation that may not yet exist in the population, or which might take longer when only using crossover. This random introduction of variation into the population not only works to help expand and further explore the program's search space, but also helps to offset search bias that may have been introduced into the population when it was first initialised (D'backhaeseleer 1994).

The following paragraphs describe a number of different ways in which mutation can be used in GP environments. It has been shown that mutation is not necessarily a key evolutionary component in GP, and that crossover alone can be sufficient (Koza 1994, Koza 1992) for generating solutions. However, mutation can optimise a tree in ways that crossover cannot, by adding diversity to the population, and helping explore the program's search space more effectively (Harries & Smith 1997). Research suggests that mutation can work without the assistance of crossover (*Evolving Computer Programs without Subtree Crossover* 1997). It has also been shown that more than one of these mutation operators can assist in improving the performance of the population (Kraft, Petry, Buckles & Sadasivan 1994, Angeline 1996). Harries and Smith (1997) were also able to demonstrate that mutation alone can outperform crossover-only solutions.

Subtree mutation finds a randomly-selected subtree, then generates another randomly selected subtree to replace it (Koza 1992). A slight change to this approach is to limit the depth of the subtree selected, so that the newly generated subtree is no more than 15% deeper than the parent tree (Kinnear, Jr. 1993).

Size-fair subtree mutation (Langdon 1998) finds a random subtree and attempts to replace the tree with another randomly-generated subtree of the same or similar size. The size of the new subtree could be in the range of 50%-150% of the size of the subtree being replaced.

Node replacement mutation works by simply taking a single, randomly-selected node and randomly changing it to guard against corruption of the tree structure. The node is checked to ensure that it uses the same number of arguments as the node that it replaces (McKay, Willis & Barton 1995). This kind of mutation functions similarly to linear genetic algorithms, where a random position in the program has its bit flipped (Bäck, Graaf, Kok & Kosters 2001).

Hoist mutation creates a new offspring by randomly selecting a subtree from its parent. The root node is then set at the base of this selected subtree. This results in the new offspring always being smaller than the parent.

Shrink mutation randomly selects a subtree, and replaces this node with a terminal (Angeline 1996), effectively reducing the overall size of the tree. Much like hoist mutation, it acts to reduce program size.

Permutation mutation randomly chooses a function node, and then randomly permutes its arguments. This type of mutation can sometimes be ineffective, as Koza demonstrates (1992). However, Maxwell (1996) used a similar approach called "swap", with restrictions to affect only binary and non-commutative functions.

Random constant mutation (Schönauer, Sebag, Jouve, Lamy & Maitournam 1996) works by mutating the constants within the tree, to try and improve the tree's overall fitness. A single mutation is defined as a single change to a constant within the tree. Systematic constant mutation goes one step further by optimising trees. This expensive method acts to find the "best" constant values. Nikolaev & Iba (2006) have used this approach to optimise each tree modified by crossover.

2.5 Evolving Interest Point Detectors

This section consists of an overview of interest point operators and its application in the field of EL, with a primary focus on GP.

Given the challenges of finding optimal parameters for reliable feature detection, the field of EL has endeavoured to optimise feature detection in a variety of automated manners, including EL approaches based on PSO (Cagnoni, Dobrzeniecki, Poli & Yanch 1999, Owechko & Medasani 2005, Cagnoni, Mordonini & Sartori 2007, Gao, Xu, Sun & Tang 2010). This aspect of evolution has also expanded to include 3D points and colour histogram matching as part of its fitness function (Cagnoni 2014), a reflection of more conventional 3D interest point detectors mentioned in 2.2.5. Genetic algorithms (GA) have also been used to optimise feature extraction (Trujillo, Legrand, Olague & Pérez 2010).

2.6 Genetic Programming in computer vision

Tackett, (1993) and later Ebner (1997, 1998), were the first to initially approach the problem domain of image feature classification optimisation, recognising that it could be seen as an optimisation problem that could be addressed by GP. Other early applications of GP to address the problem domain of feature detections were conducted by Poli and involved image segmentation (1996*b*, 2009). Other later work addressed basic object (Smart & Zhang 2003) and texture classification (Song, Loveard & Ciesielski 2001, Song, Ciesielski & Williams 2002, Song & Ciesielski 2008). Approaches to optimising camera placement to assist in applications such as photogrammetry (Olague & Mohr 2002) were conducted based on a virtualised 3D environment, as well as the utilisation of "honey bee swarms" of 3D points (Olague & Puente 2006). This involved 2D points from different viewpoints intelligently triangulating to form a 3D point cloud in Euclidean space. Lin and Bhanu (2005) also utilised GP for object recognition in Synthetic Aperture Radar (SAR) images.

Some of the earliest work in this field, where well-established CV feature detectors could be made compatible with GP, can be attributed to Olague and Trujillo (2006), (2006), who pioneered the concept of fitness metrics that shared evaluation characteristics that were well accepted among the CV community (Schmid et al. 2000, Mikolajczyk & Schmid 2001). Olague and Trujillo's utilisation of evaluation from the CV field helped normalise evaluation of GP classifiers, with subsequent work establishing single and multi-objective applications (G. & L. 2011). Many others have built on this methodology, and adopted strategies to evolve IP detector invariance using the F-measure (Perez & Olague 2008, Perez & Olague 2009*b*, Perez & Olague 2009*a*), a hill-climbing optimisation (Perez & Olague 2013). Others have largely reused their algorithms for colour-based fusion of features (Shao et al. 2014). It has also been adapter for edge-based feature detection by leveraging Gaussians in conjunction with the F-measure (Fu, Johnston & Zhang 2011, Fu, Johnston & Zhang 2013, Fu, Johnston & Zhang 2016). It has also been used in conjunction with Histogram of Gradients (HoG) (Lensen, Al-Sahaf, Zhang & Xue 2015, Lensen, Al-Sahaf, Zhang & Xue 2016). Even more recent work to optimise rotation invariance (Al-Sahaf, Al-Sahaf, Xue, Johnston & Zhang 2017) still shares a common connection with Olague and Trujillo's earlier research. In all these cases, their initial research in GP and CV has formed the basis for this more recent research in one way or another.

It is difficult on the surface to observe how a sophisticated image filter of this type can be used in a genetic programming design (as described initially in Section 2.2.6), as it requires uniformity between functions so that, regardless of where the functions lie in the algorithm, it can function. This type consistency is necessary for crossover, as well as for ensuring that there are no special corner cases where the evaluation of the individual will fail. However, the nature of image processing makes it somewhat simplified when it is taken into consideration that the images are processed as a matrix. This means that the consistency of inputs and outputs can be ensured and concerns over evaluation safety can be largely eradicated, as long as other conventions regarding GP processing are enforced.

Because every input for the functions in the GP tree involves the processing of the entire image, the functions used can be interchangeable. Thus, it doesn't matter when or how they are used, as they will always use data derived from the original image, which were defined as terminals, or data that has already been pre-processed by other functions. Processing images as a matrix allows the entire algorithm to be represented as a tree, with interchangeable nodes, which are better known as functions, and terminals that each function processes. This enables the re-representation of interest point filters, so that they can be easily manipulated as an inverted tree. The Harris algorithm, therefore, can be represented as:

$$\begin{aligned} K = -(-(*(G_{\sigma=2}(I^2_{out}(L_x)), G_{\sigma=2}(I^2_{out}(L_y)), (I^2_{out}(G_{\sigma=2}(*(L_x, L_y)))), \\ k \cdot I_{out}(+((G_{\sigma=2}((I^2_{out}(L_x)), G_{\sigma=2}(I^2_{out}(L_y))))) \end{aligned} \tag{2.37}$$

Equation (2.37) redefines the Harris algorithm in a way that still processes the image in exactly the same manner, but represents it in prefix notation. Prefix notation better illustrates the flow of processing in a manner that enables the functions being performed on the image to be properly delineated, as well as abstracting away window processing and the matrix processing being performed, as they are implied to be occuring at all levels. To sufficiently describe the processing involved, the processes are classified into their respective functions, and terminals. In Harris, we have the functions:

$$F = \Big\{+, -, *, I^2_{out}, k \cdot I_{out}, G_{\sigma=1}, G_{\sigma=2}\Big\}$$

as well as the terminals,

$$T = \Big\{L_x, L_y\Big\}$$

where L_u are already pre-processed Gaussian derivatives $\frac{\delta}{\delta U} G_{\sigma_D}$ of the image I. I_{out} is used to represent the fact that the output of a lower subtree is required as an input, or requires a single terminal T. This prefix notation representation means it is effortless to move, add to, or even remove subtrees within the algorithm for the purposes of mutation and crossover, to enable evolution of an image filter.

Figure 2.3 is an illustration of the internal image data at each stage of processing within the interest point detector with a false colour intensity gradient. From the terminals labeled as L_x, L_y, L_{yy} and I_r (I_r being the intensity image of the red channel), the image terminals pass up the tree via each function, and are

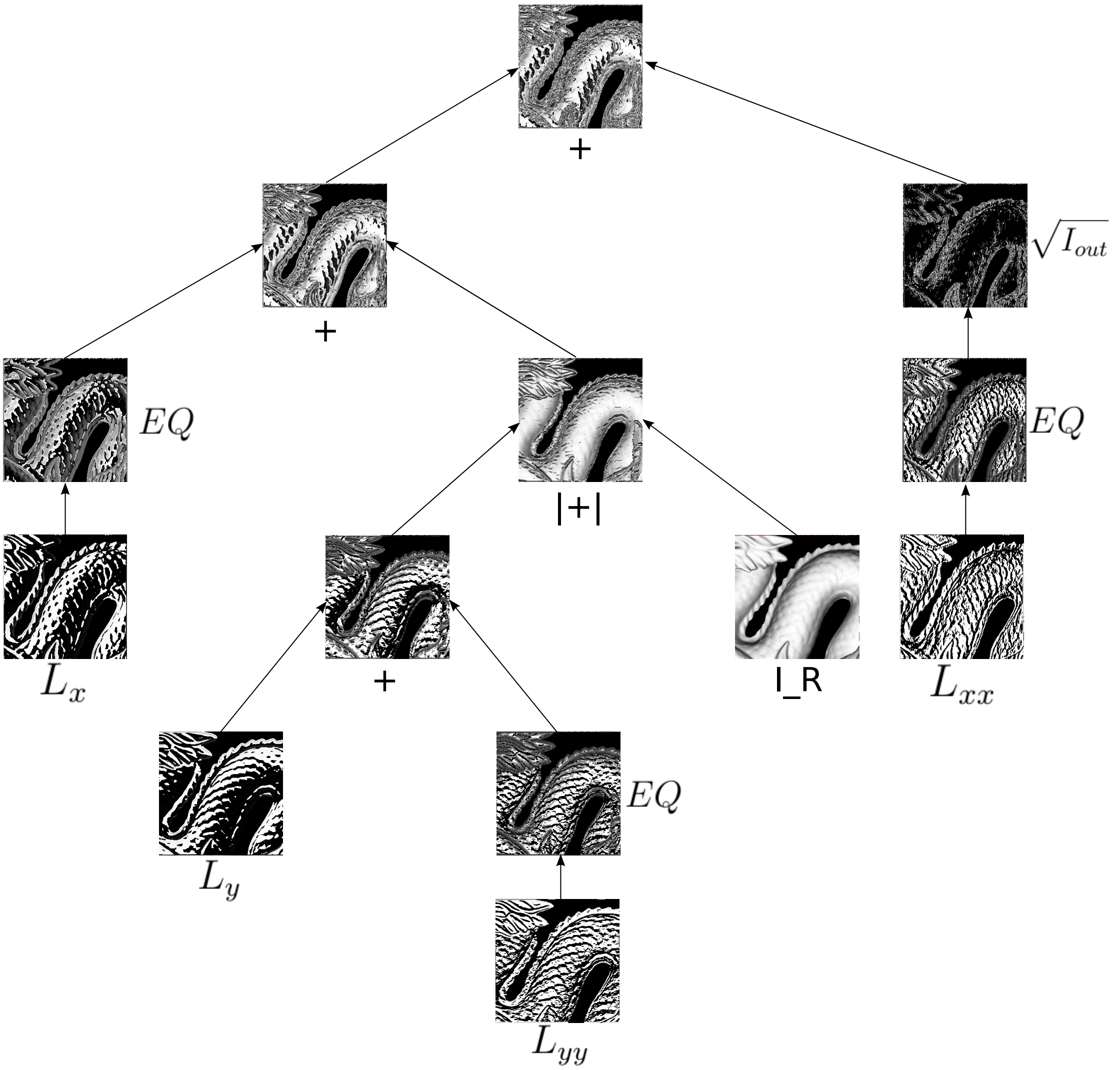


Figure 2.3: Example of a syntax tree representing how a detector can be described using GP trees.

recombined. This recursive processing results in the final response image at the top of the inverse tree. The image at the apex of the tree is then thresholded to find the points of interest. Because each image is processed as a whole, this makes it extremely easy to keep the inputs to the functions consistent.

2.7 Interest Point Detectors in GP

This section will outline how detectors can be broken down into tree structures that can be easily manipulated using GP crossover and mutation without affecting functionality. It will also describe existing research that has successfully identified methods for codifying the structure of image interest point detectors and implementing a fitness function that can effectively measure detector performance, as well as identifying and addressing undesirable biases that evolution can introduce into an evolved detector.

Based on Olague and Trujillo's earlier work (Trujillo & Olague 2008) they adapted it (2011), but with a multiobjective approach (Coello et al. 2007, Zhou, Qu, Li, Zhao, Suganthan & Zhang 2011), in both cases however, their work measured the salience of points from a detector via repeatability done by Schmid (2000) and though it has been adapted in other research, assessing the repeatability of interest points has remained static. For the purposes of this dissertation, this section will only be focusing on the earlier research, which utilised a fitness function that assessed repeatability of points, point dispersion in the scene, and applying a fitness penalty proportional to the number of returned points.

2.8 GP Fitness Function

From the measures of repeatability $r_{K,J}$, point dispersion ϕ_u, and penalty factor $N^{\%}$, the fitness of a candidate detector can be effectively quantified. In addition to this Olague and Trujillo also used static values α, β, γ, a_u and c_u, based on empirical analysis, to improve the fitness values. The fitness function is represented as a sequence of multiplied layers.

$$f(K) = r_{K,J}(\epsilon) \cdot \phi_x^{\alpha} \cdot \phi_y^{\beta} \cdot N_{\%}^{\gamma} \tag{2.38}$$

With $r_{K,J}(\epsilon)$ representing the Schmid repeatability of all the scenes being tested, and ϕ_u representing the point dispersion of the points in the scene for the x and y axis of the reference transform image,

$$\phi_u = \begin{cases} \frac{1}{1+e^{-a(H_u - c)}}, \text{ when } H_u < H_u^{max}, \\ 0 \text{ otherwise} \end{cases} \tag{2.39}$$

with H_u^{max} for the x and y axis being empirically determined.

$$H_u = -\sum P_j(u) log_2[P_j(u)], \tag{2.40}$$

H_u in equation (2.40) is the entropy value of the detected points along direction u and is calculated based on the reference image I_1 from J_1 only. This only gives the fitness function a small snapshot of the point dispersion across all of the scenes in

J. $P(\cdot)$ represents a histogram of points over 8 bins.

The last element of the fitness function,

$$N_{\%} = \left(\frac{\text{extracted points}}{\text{requested points}}\right)^{\gamma} \tag{2.41}$$

is a penalty to the fitness value if the detector returns less than the number of points desired.

2.8.1 Point Dispersion

Measuring repeatability on its own in a GP context provides additional challenges as the tendency of detectors to converge on solutions that are not not desired can be complex, and unpredictable. If a fitness function is not designed to avoid certain optimisation strategies, in simple terms, the algorithm may "cheat". To mitigate, or at least offset these situations, other characteristics of the detector are also measured or controlled. Being able to filter for point dispersion of interest points in a scene can help prevent clustering, or clumping of points. Point dispersion is calculated with a 2D histogram, H, by finding the Shannon entropy (Shannon & Weaver 1949) of the x and y positions u of points P found within I, from the scene j (in this instance j is always the reference transform) and is shown in equation (2.40). The relevant histogram is then used in a sigmoidal function (2.39).

2.8.2 Number of Points Detected

Another notable metric that can be exploited by the evolutionary process is the number of points returned. To that end, a penalty factor is used to reduce the overall fitness value proportional to the number of points returned. This ensures that the evolutionary process does not exploit loopholes, such as returning fewer points in order to push up the repeatability performance, or exploiting point dispersion by making it easier to cluster points in a less surreptitious manner. The percentage of actual extracted points and the maximum requested points can then be applied to the resulting fitness score as a penalty to ensure that evolved detectors return as many points as possible, as shown in equation (2.41).

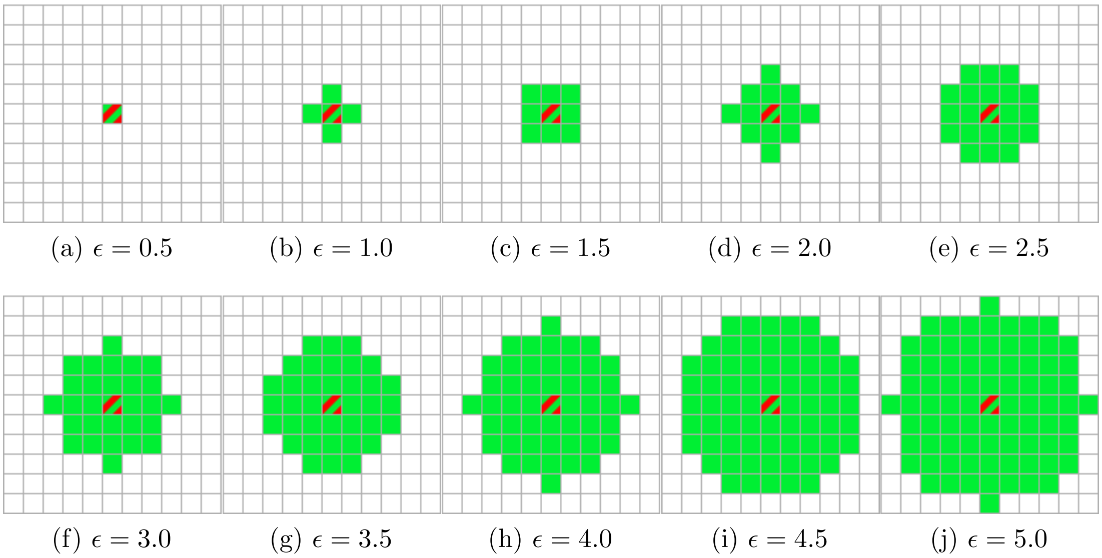
2.9 Classification and Evaluation

This section explores some of the more well-known classification and evaluation methodologies used in measuring the performance of keypoints, and, to a more limited extent, image descriptors produced by 2D-based feature detectors. It aims to cover two major areas in performance evaluation, where the somewhat arbitrary classification of features in CV meets the more rigid statistical analysis of the field of EC, or, more specifically, GP for keypoint optimisation. It serves as a more focused review of existing classification and evaluation in CV and GP respectively, as well as highlighting some of the flaws and limitations of current approaches. This section also serves as the inspiration for the major contributions highlighted in Chapter 1.

2.9.1 Keypoint Classification via Repeatability

In Section 2.2.1, a very brief overview of evaluation performance was made. It was shown interest points usually fall under two broad categories, repeatability, and information content. In a general sense, based on the needs of the computer vision field, keypoint performance has been mostly relegated to repeatability. With newer image descriptors, other types of performance evaluation like orientation can be taken into consideration, but evaluation of extra information beyond point position is generally seen as a secondary priority.

Based on the need to track features, it should be generally accepted that repeatability is a major benchmark for measuring detector performance. Repeatability is also easy to measure and a common metric to compare, as points can be easily quantified and compared across candidate detectors that are designed to discover points of interest in a scene. Schmid's (2000) measuring of repeatability has been extremely popular over the years when trying to compare the performance of detectors across scenes, as it can accurately quantify general performance. When determining the repeatability of a detector, $r_{K,J(\epsilon)}$ represents the overall repeatability performance of the detector K when it is run on each scene J that contains the transformed images I_i, where $i = 1 \ldots N$. The initial image I_1 is set as the reference image in the scene, and is used to compare to the repeatability of every other image that results from the scene being transformed. From this, we



(a) $\epsilon = 0.5$ (b) $\epsilon = 1.0$ (c) $\epsilon = 1.5$ (d) $\epsilon = 2.0$ (e) $\epsilon = 2.5$

(f) $\epsilon = 3.0$ (g) $\epsilon = 3.5$ (h) $\epsilon = 4.0$ (i) $\epsilon = 4.5$ (j) $\epsilon = 5.0$

Figure 2.4: ϵ from 0.5 to 5.0. The central point represents the reference point, denoted as striped red and green, and the green areas denote valid areas for a candidate point to be classified "repeated".

get the average repeatability of all the scenes,

$$r_{K,J(\epsilon)} = \frac{1}{N-1}\sum_{2}^{N} r_{K,I_i(\epsilon)} \tag{2.42}$$

which returns a value from 0 to 1 across all scenes.

Building on top of this work, a methodology for repeatability was created to measure scale-invariant interest points (Mikolajczyk & Schmid 2001), described as:

$$r_{1,2} = \frac{C(I_1, I_2)}{mean(m1, m2)} \tag{2.43}$$

In this case $C(I_1, I_2)$ represented the number of corresponding couples, and $m1, m2$ represented the numbers of detected points in each of the images. A repeated, or "corresponding" pair were considered repeated if the error (the distance between the two closest points) was less than 1.5 pixels, represented as:

$$C(I_1, I_2) = |x_a, H \cdot x_b| < 1.5 \tag{2.44}$$

(which corresponds to an ϵ of 1.5, and, in addition, the ratio and detections at varying scales didn't differ from it's real scale ratio by 20%. This mirrored the same basic principle of a ratio of true positives and true negatives, and an error threshold (ϵ) of 1.5. Additional research for testing scale invariance (Mikolajczyk

& Schmid 2002) also used a similar testing methodology based on a homography of points to measure distances, ratios of true positives, and true negatives, and an ϵ of 1.5.)

Other work by Schmid and Mikolajczyk (Mikolajczyk & Schmid 2005, Mikolajczyk, Tuytelaars, Schmid, Zisserman, Matas, Schaffalitzky, Kadir & Gool 2005) approached the problem of repeatability based on a recall and the (1-precision) metric in a more refined manner that was more applicable to areas, or blobs. Their approach took a pair of regions A and B and considered their descriptors D_A and D_B matched if the distance d were below a certain threshold t, from which recall and (1-precision) of the scene could be determined, as well as the number of correct matches (those falling below the threshold), correspondences (possible correct matches) and false matches (those above the threshold). Recall was defined as the number of correctly-matched regions between two images taken from the same scene, but under differing conditions:

$$recall = \frac{\#correctmatches}{\#correspondences} \tag{2.45}$$

and precision,

$$1 - precision = \frac{\#falsematches}{\#correctmatches + \#falsematches} \tag{2.46}$$

The correct matches and correspondences were determined based on an overlap error (Mikolajczyk et al. 2005), represented as:

$$1 - \frac{R_{\mu_\alpha} \cap R_{(H^T \mu_\beta H)}}{(R_{\mu_\alpha} \cup R_{H^T \mu_\beta H})} < \epsilon 0 \tag{2.47}$$

with R_μ representing the elliptic region, defined as $x^T \mu x = 1$, and H being the

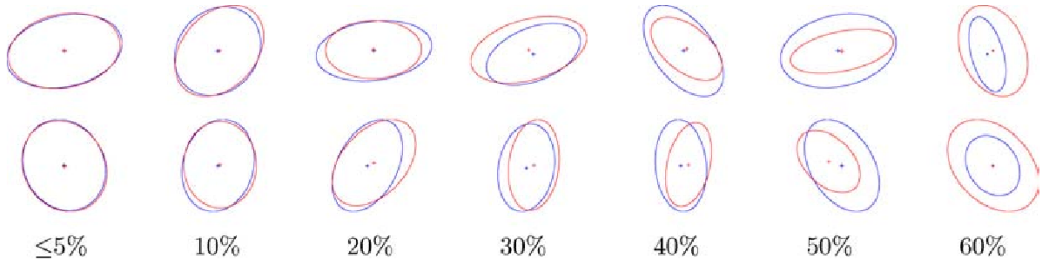


Figure 2.5: Examples of image descriptor overlap percentages, from Mikolajczyk et al. (2005). Reused under Fair Dealing for Criticism or Review.

homography of the two related images. $R_{\mu_\alpha} \cap R_{(H^T \mu_\beta H)}$ and $(R_{\mu_\alpha} \cup R_{H^T \mu_\beta H})$ each represented the union and intersection of these regions, and $\epsilon 0$ in this instance was set to an arbitrary value of 40% ($\epsilon 0 < 0.4$), see Figure 2.5. This resulted in a repeatability ratio of the number of region-to-region correspondences within the image pair, and produced a ratio, however only areas of the scenes present in both images are computed. As can be seen in this instance, the repeatability metric is somewhat different to previous work in that it relates to areas, rather than points. However much of the fundamental approach incorporates much of the previous work, for example, the ϵ threshold for classifying a repeated point, or in this case are repeated, as well as the use of a homography to compare the detections from two scenes, and a ratio of repeated, and not repeated detections, based on the use of a reference scene.

Each of these methodologies still share a common core set of evaluation aspects:

- They utilises 2D image data, and rely on marking up image sets, and calibration of image orientation.
- A homography is required to transform the points detected based on the scene orientation.
- Keypoints use an error rate, or ϵ, of 1.5 to classify a point as "repeated" or not.
- A reference scene is used to compare other candidate points.
- Only points that can be seen from both camera viewpoints are included in the evaluation.

It is worth nothing that though image descriptors have a more complex repeatability methodology, an image descriptor's keypoint is still required to pass the 1.5ϵ threshold first before further evaluation (Mikolajczyk & Schmid 2005, Mikolajczyk et al. 2005). This core requirement has not changed since Schmid's original work on interest point repeatability (Schmid et al. 2000), but Schmid and Mikolajczyk's work on performance evaluation since then is, even today, still very heavily used.

2.9.2 Interest Point Detector Performance

Current feature detectors, though effective and varied, are continually undergoing improvements to increase performance across a multitude of different applications.

One of the most demanding and sought-after performance metrics that undergoes refinement is repeatability of detected features. Whether interest points are keypoints, or image descriptors, repeatability of detected features is a metric that is extremely relevant in most areas of computer vision, and across the whole field of pattern recognition. The robustness of detected features when challenged with various transformations, (especially affine transformations to the scene like rotation, scale, translation and shape), noise, illumination etc., is an ongoing research subject that often focuses on the repeatability of such features (Azad et al. 2009, Mikolajczyk & Schmid 2004).

Though there has been a great deal of research on creating variations on keypoints, image descriptors and classifiers in general, the general methodology for evaluation has changed little. Most of the literature cited thus far evaluates repeatability based solely on the work initially established, and built on by Mikolajczyk, Schmid, et al. (Schmid et al. 2000, Mikolajczyk & Schmid 2001, Mikolajczyk & Schmid 2004, Mikolajczyk & Schmid 2005). This work essentially uses the ratio of repeated points versus the maximum of either the reference scene, or scene being compared, or the recall (number of correct matches / number of correspondences) vs 1-precision curve (number of false matches / number of matches), of detected features (Mikolajczyk & Schmid 2005). In simplified terms these repeatability approaches use true positives and true negatives to calculate a repeatability curve. Though it has been of great benefit to have a consistent methodology for researchers, there are clear limitations to this performance metric that will be further outlined at the end of this chapter. This type of approach to repeatability that is used in other research covered in this literature review only includes the evaluation of true positives and true negatives, as well as a single set of reference points for comparison. In nearly all cases, this application of repeatability uses a homography of those points, which substantially limits the available scene transformations that a set of 2D points can be subjected to.

2.10 Evaluation Methodologies of Classified Features in GP and CV

This section aims to cover some of the major evaluation techniques which, over the years, have been utilised in computer vision feature classification and evaluation. It first outlines the more general evaluation concepts that are applicable to CV, GP and statistics, and then moves to identify the currently-used eval-

	+R	**-R**	
+P	TP	FP	pp
-P	FN	TN	pn
	rp	rn	N

Figure 2.6: Contingency matrix. Green cells represent correct classification, and red cells, incorrect.

uation techniques that have been adopted. The generally accepted convention for classification in statistics is a confusion matrix that delineates the positive and negative predicted classifications, in contrast to known positive and negative clasifications. The two main overarching classification categories are the predicted, and actual classifications. Each of these consist of positive and negative cases. Predicted and true, is represented as $\mathbf{P}+$, predicted but false as $\mathbf{P}-$, actual and true as $\mathbf{R}+$ (also referred to as recall, or real classifications), and actual but false as $\mathbf{R}-$. From these predicted and real classifications, the data sets of predicted true and false classifications $\mathbf{P}$ can be compared to already-known real classifications $\mathbf{R}$ to determine the performance of a classifier. The intersections of $\mathbf{P}$ and $\mathbf{R}$ on the contingency table result in the classifications of true positives ($+\mathbf{P}$ and $+\mathbf{R}$), true negatives ($-\mathbf{P}$ and $-\mathbf{R}$), false negatives ($-\mathbf{P}$ and $+\mathbf{R}$), and false positives ($+\mathbf{P}$ and $-\mathbf{R}$). The conventions of upper case contingency table terms have been adopted to represent counts or the sum of classifications, and lower case contingency table terms to represent ratios or proportions in relation to the total classification cases within the confusion matrix, represented as N. From these definitions, the predicted positives ($\mathtt{pp} = \mathtt{TP} + \mathtt{FP}$), predicted negatives ($\mathtt{pn} = \mathtt{FN} + \mathtt{TN}$), real positives ($\mathtt{rp} = \mathtt{TP} + \mathtt{FN}$), and real negatives ($\mathtt{rn} = \mathtt{FP} + \mathtt{TN}$) can be identified. Finally, N represents the total of all possible classifications in the contingency table ($N = \mathtt{TP} + \mathtt{FP} + \mathtt{TN} + \mathtt{FN}$). The contingency table layout, shown in figure 2.6 forms the basis of the types of evaluation within CV, GP, and this dissertation. Equations 2.48 to 2.55 illustrate some of the major metrics used, and will be referred to in the following sections.

$$\begin{aligned}\text{Recall} &= \text{Sensitivity} = \texttt{tpr} = \texttt{tp/rp} \\ &= \texttt{TP/RP} = \texttt{TP/(TP} + \texttt{FN)}\end{aligned} \tag{2.48}$$

$$\begin{aligned}\text{Precision} &= \texttt{tpa} = \texttt{tp/pp} \\ &= \texttt{TP/PP} = \texttt{TP/(TP} + \texttt{FP)}\end{aligned} \tag{2.49}$$

$$\begin{aligned}\text{Inverse recall} &= \texttt{tnr} = \texttt{tn/rn} \\ &= \texttt{TN/RN} = \texttt{TN/(FP} + \texttt{TN)}\end{aligned} \tag{2.50}$$

$$\begin{aligned}\text{Inverse precision} &= \texttt{tna} = \texttt{tn/pn} \\ &= \texttt{TN/PN} = \texttt{TN/(FN} + \texttt{TN)}\end{aligned} \tag{2.51}$$

$$\begin{aligned}\text{Accuracy} &= \texttt{tp} + \texttt{tn} \\ &= \texttt{rp} * \texttt{tpr} + \texttt{rn} * \texttt{tnr} \\ &= \texttt{pp} * \texttt{tpa} + \texttt{pn} * \texttt{tna} \\ &= \texttt{TP} + \texttt{TN} = (\texttt{TP} + \texttt{TN})/\texttt{N}\end{aligned} \tag{2.52}$$

$$\begin{aligned}\text{Fallout} &= \texttt{fpr} = \texttt{fp/rn} \\ &= 1 - \text{Inverse recall} \\ &= \texttt{FP/RN} = \texttt{FP/(FP} + \texttt{TN)}\end{aligned} \tag{2.53}$$

$$\begin{aligned}\text{Specificity} &= \texttt{fpr} = \texttt{tn/rn} \\ &= \texttt{TN/RN} = \texttt{TN/(FP} + \texttt{TN)}\end{aligned} \tag{2.54}$$

$$\begin{aligned}\text{Miss rate} &= \texttt{fnr} = \texttt{fn/rp} \\ &= \texttt{FN/RP} = \texttt{FN/(FN} + \texttt{TP)}\end{aligned} \tag{2.55}$$

Image feature classification and evaluation with GP has been explored for well over two decades now, and has gone through a series of changes in approach, and implementation. However, all classificaton and analysis of those classifications use some form, or combination of the above to define fitness based on their particular requirements, though arguably, most tend to focus on true positive classification, or more broadly, recall and precision, and tend to omit true negative classifications entirely for reasons that will be discussed in Section 2.10.1.

2.10.1 Evaluation Basic Concepts

Illustrated in equations 2.48, 2.49, 2.50 and 2.51, are the basic evaluation metrics of classification systems. recall and precision focus on the true positives of the predicted, and real classifications. To coin a simple example of precision and recall, consider a situation in which you are asked to list 5 different types of mammals. In this scenario, you aren't sure whether the animals you have in mind are indeed mammals, and so you list the names of 7 animals instead, in the hope that 5 are mammals. After listing the names of 7 animals, you check your responses against a list of correct answers, and confirm that 5 of the 7 animals you listed were indeed mammals. This means that your recall 5/(5+0)=1.0 was perfect (100%), but your precision was only 5/(2+5)=0.714 (74.1%). While recall only focuses on true positives (i.e., in the example, it measures the extent to which you guess the correct mammal names), the precision catches the incorrect guesses, and will continue to get lower as more incorrect guesses are made, whereas recall only focuses on checking whether the total guesses has been accounted for. The emphasis on precision helps to avoid allowing "faking it till you make it" classifiers making more attempts (which becomes closer and closer to guessing) than other classifiers that may not get every classification correct (which may result in lower recall), but still manage to get most classifications correct with less attempts.

Precision's inclusion of false positives in this way, (also known as type I errors), plays a strong factor in many classification strategies, as will be covered later. The inverse of recall, referred to as the true negative rate (`tnr`) takes the proportion of true negatives compared to the real negatives (2.50), and for inverse Precision, also known as the true negative accuracy (`tna`), takes the proportion of true negatives compared to the predicted negatives (2.51). Inverse precision covers the instance of false negatives, which are referred to as type II errors. Type I and II errors (that cover false positives, and false negatives respectively), can be succinctly described as false hits, and false misses, in which one case registers a positive classification, but gets it wrong, and the other fails to register a negative classification when it should have, hence the "miss".

To give a real example, a type I error would be similar to taking a pregnancy test and the pregnancy test (the classifier) determining that the person is pregnant when they are not, making the pregnancy test result a false positive. In the case of a type II error, if the same test determined the person was not pregnant when they are, that would be a false negative. Given the consequences of these two error types, many consider a type I (or false positive) reading to be more misleading than a type two, even though, as seen in both examples, a misclassification has the same consequences, the only difference being that a type I allows the incorrect rejection of the null hypothesis, whereas the type II does not, and means the null hypothesis can't be determined.

Accuracy serves to take into consideration correctly classified negatives, and can be expressed as a weighted average of precision and inverse precision, and as the weighted average of recall and inverse recall (2.52). Compared to the recall of true positives, the fallout, or false positive rate `fpr`, measures the prevalence of misclassifications proportionate to the real negatives (2.53). Lastly, the miss rate (2.55), as its name implies, measures the proportion of missed classifications.

From the contingency table in figure 2.6, recall (also known as the true positive rate, or *tpr*) can be represented as $\texttt{Rec} = \texttt{tpr} = \texttt{tp}/\texttt{rp}$, and precision (also known as the true positive accuracy, or `tpa`) can be represented as $\texttt{Prec} = \texttt{tpa} = \texttt{tp}/\texttt{pp}$.

In some of the earliest work on feature classification for GP, Poli used the sum of false positives and false negatives of incorrectly classified pixels (which was somewhat inferred and not properly ascertained based on the true positives and true negatives), which Poli attempts to minimise using a weighted threshold α (Poli 1996*c*). Ebner used the squared pixel distances of the actual and the desired points compared to an existing operator (true positives only), (Ebner et al. 1998). Some GP evaluation for image segmentation-based GP used dynamic range selection and a ratio of true positives and true negatives (accuracy) (Song et al. 2002, Smart & Zhang 2003). More recent work use ratios of repeated and non-repeated keypoints (true positives and false positives) as part of the fitness evaluation of a classifier (Gustavo & Leonardo 2006, Trujillo & Olague 2008, G. & L. 2011).

In CV, there is a clear focus on recall and precision, especially for keypoint evaluation, as shown in Section 2.9.1, or more specifically equations 2.42, 2.44 and 2.47 as a means of gauging classification. However, the real classifications are the reference keypoints/interest points, and the predicted classifications are the other scenes it is compared to. As will be pointed out near the end of this section, this creates a bit of a misnomer with regard to what is classified real, and what is predicted with relation to the ground truth that the classifier uses. This is especially aparrent with the optimisation strategies in GP, where the real classifications are dependent on a reliable ground truth, or dataset, on which training is conducted.

2.10.1.1 Receiver Operator Characteristics (ROC)

Receiver Operator Characteristics (ROC), is a particularly popular statistical measure that has been utilised across many fields since as early as WW2

(Woodward 1953). ROC is capable of the representation of binary classification systems across multiple classification thresholds. Unlike recall and precision, ROC measures the ratio of the recall, or true positive rate (`tpr`), compared to the fallout, or false positive rate (`fpr`). These are also referred to as sensitivity, and 1-specificity respectively. This ratio is also normalised (or scaled) regardless of bias, and can be represented as a curve, also known as a ROC curve. Figure 2.7 illustrates the types of data points that can be expressed on a ROC graph. The areas of the ROC illustrate a number of different tradeoffs for the classifier being evaluated. In an ideal situation, "D"$= (0, 1)$ is best, where true positives are maximised and false positives are minimal. Points such as "A"$\approx (0.6, 0.1)$ would be considered a more conservative assessment than "B"$\approx (0.8, 0.5)$, even though B has better classification. Data points like "B" that move closer to the upper right could be considered more "liberal" comparatively, and, depending on the classifier, could still be considered acceptable, but their increased true positive classification comes at the cost of increased false classifications. "C"$\approx (0.7, 0.7)$, on the other hand, is essentially random and could be considered the worst area on the graph. Points that rest on the chance line (in figure 2.7 this is represented as the dashed line), highlight that the classifier's performance is no better than chance. "E"$\approx (0.65, 0.22)$ performs even worse than chance, also referred to as "perverse", represents that classification performance is inverse to the desired behaviour. Ironically, this performance data point could be considered better than chance, but it also indicates that the classifier is functioning in an undesirable manner. To coin an example, a "perverse" classifier could be likened to a sprinter running in the opposite direction around a running track, and even though this could be considered "wrong" it is still fast, and the time is still valid. In most cases data points appear above the chance line rather than below, but even for data points appearing below, they are still performing some degree of correct clasification.

Because this ROC curve includes all possible classification thresholds, it provides advantages such as finding the tradeoffs at each classification threshold without needing to "juggle" or balance the position of best fit for the `tpr/fpr` tradeoff, as well as being able to calculate the sum of the tradeoffs as a single metric, more commonly referred to as the Area Under the Curve (AUC) (2.56). Because the curves are normalised across (0, 1), this also makes it easier to compare different datasets. The advantages of ROC and AUC can also be seen as it incorporates type I and type II errors but this does not account for "perverse" classifier behaviour however as it treats the `tpr` and `fpr` tradeoff as

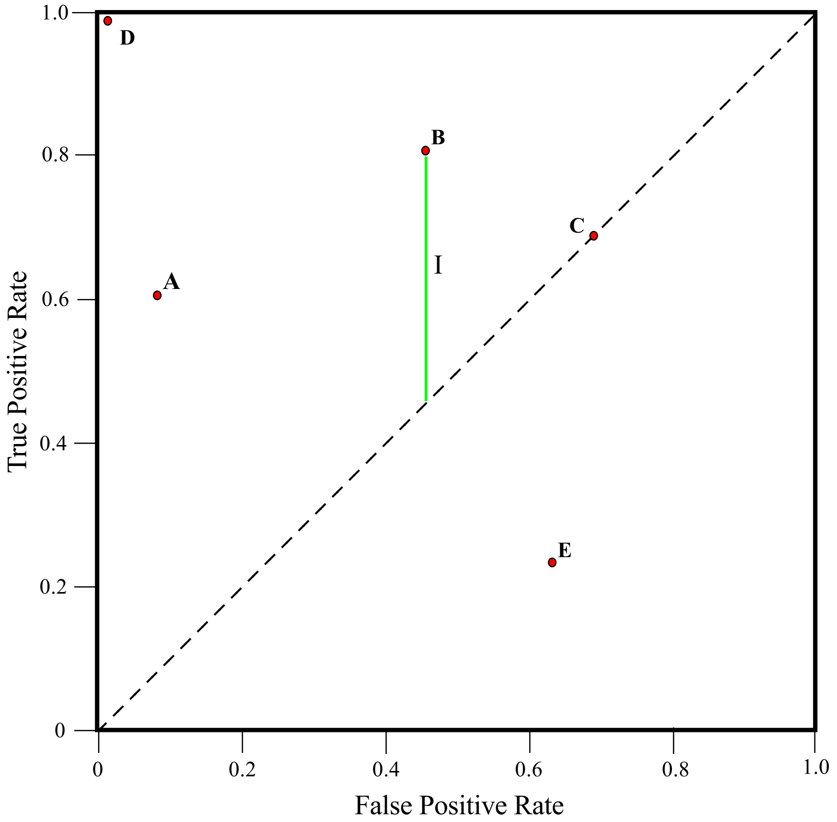


Figure 2.7: ROC graph showing the five types of classification data points (A, B, C, D, E), and Informedness (I).

equally important, and though the AUC can be used as an overall metric, neither ROC nor AUC on their own are sufficient for comparing different datasets at specific thresholds, nor can they accurately provide metrics that can determine the threshold for the best fit tradeoff. They are, however, often used when the overall predictiveness of a classifier is required. This in itself can be misleading if variance is not taken into consideration, and can be problematic if the `fpr` is not fixed or the variance is in more than one dimension (Fawcett 2004). The AUC for a specific classifier is closely related to the informedness (covered in Section 2.10.2) of the classifier:

$$\begin{aligned} \text{AUC} &= (\text{Sensitivity} + \text{Specificity})/2 \qquad (2.56) \\ &= (\text{Informedness} + 1)/2 \qquad (2.57) \end{aligned}$$

There is, however, a degree of doubt over the benefits of AUC, due to its inability to properly consider cost distributions between classifiers (Hand 2009). In the case of evaluation of interest points that uses a threshold, as discussed earlier in Section 2.9, the use of ROC, or AUC is not necessarily appropriate due to the way in which classifiers are compared at specific thresholds. In the cases of keypoints and image descriptors, a single threshold is utilised, which makes ROC/AUC evaluation ineligible. The ROC/AUC performance of features has been measured in other types of feature evaluation however, particularly machine learning, with varying degrees of success (Viola & Jones 2004, Schiele & Kruppa 2003).

With that in mind, ROC curves have been used in GP to evaluate features (Winkeler & Manjunath 1997, Zhang & Rockett 2005), (though Zhang et al. admit their classification process is somewhat crude), as well as other edge detection using ROC analysis (Bowyer, Kranenburg & Dougherty 2001, Wright, Ma, Mairal, Sapiro, Huang & Yan 2010). Other work has also examined various types of fitness functions in GP, and demonstrated that AUC metrics converge on optimisations faster and perform better overall (Bhowan, Johnston & Zhang 2012). Interestingly, some studies using genetic algorithms found that ROC and informedness are particularly useful, as discussed later in Section 2.10.2.

However one of the main arguments against ROC is that Real Negatives are difficult to properly quantify but this essentially just a scale constant (Powers 2011), and the same can be said for Real Positives as the number of interesting points or features is difficult to know.

2.10.1.2 F-Measure

The F-Measure (Chinchor 1992, Sasaki et al. 2007), also known as the F-score (or $\mathbf{F_1}$), uses the harmonic mean of precision and recall to create a single-value measure based on the recall and precision of a classifier. There are a few variations, with the general form F being a differential weight of recall and precision, but more common uses, like F_1, uses equal weight, as shown in Equation 2.58. Unlike accuracy, which relies on a symmetry of false positives and false negatives, the F-measure most commonly uses a weighted average shown as,

$$F_1 = (\frac{recall^{-1} + precision^{-1}}{2})^{-1} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.58}$$

F-measure has received particular attention in GP over the last several years (Perez & Olague 2008, Perez & Olague 2009*b*, Fu et al. 2013, Fu et al. 2016, Al-Sahaf et al. 2017). And though there is existing work justifying a move beyond the precision only methodology described in Olague and Trujillo's work, (Agarwal

& Roth 2002), to recall and 1-precision as a means of measuring the tradeoff, similarly to a ROC curve, there is a growing body of evidence to refute the suggestion that recall and precision is statistically better. These issues will be covered in more detail in the following section. Perez (Perez & Olague 2008) also recognises that this is due to the limitations in the capacity of existing systems to properly classify true negatives. As a result, recall and 1-precision have been used as a compromise both in GP and in CV, as highlighted in Section 2.9.1.

2.10.1.3 F-Measure Limitations

Agarwal et al. (2004) deserves particular mention even though their approach to training classifiers for feature detection did not achieve the same success as other methodologies at the time like Viola-Jones (2001*b*) that developed a far more reliable feature detector. This was namely because Agarwal took a different approach to interest point repeatability compared to Schmid and Mikolajczyk's work. Of particular note is the fact that Agarwal et al. analysed performance of interest points using ROC analysis, recall, 1-precision, and the F-measure (based on a training and learning set of images) to measure performance of classifiers and to find the best trade-off. It should also be noted that Agarwal et al. did not use a homography to measure the individual point distances with a fixed threshold distance, but instead tried to match clusters of points that were divided into segments, which was far more computationally complex.

Other work on repeatability (Winder & Brown 2007) used calibrated images of scenes to triangulate the 3D position of points by measuring the Euclidean distance of the points to create a histogram comparing overlapping, and non-overlapping pairs. From this a ROC curve of true matches, and false matches can be constructed, with the area under the curve (AUC) being the repeatability metric.

Perez further refined the work of Agarwal et al. (Agarwal, Awan & Roth 2004) by applying an α metric to the F-measure (2008, 2009*a*, 2009*b*) as well as utilising a homography to evolve 2D-based interest point classifiers.

Other work in feature classification for images also recognises that the F-measure is not ideal, but it is utilised purely because of it's simplicity and ubiquity, rather than being a sound statistical metric (Herrmann, Mayer & Radig 2014) or due to limitations of the ground truth being utilised (Liang, Zhang & Browne 2015). Based on this, it can be noted that there is a general acknowledgement that F-measure has drawbacks, but F-measure is tolerated due to the limitations of the experiments being conducted and datasets used.

Powers (2015*a*) identifies several issues with the F-measure, to quote:

1. “F-measure (like Accuracy, Recall and Precision) focuses on one class only
2. “F-measure (like Accuracy, Recall and Precision) is biased to the majority class
3. “F-measure as a probability assumes the Real and Prediction distributions are identical
4. “E-measure (1-F) is not technically a metric as it does not satisfy a triangle inequality
5. “F-measures don’t average well across real classes or predicted labels or runs
6. “F-measure doesn’t in general take into account the True Negatives (TN)
7. “F-measure gives different optima from other approaches and tradeoffs.

As it applies to the existing work in CV and GP, the most worrying factor regarding the adoption of the F-measure for feature classification is bias, and the fact that it does not take into account true negatives. In the cases where the F-measure could be biased in existing CV and GP classification approaches, attempts to mitigate such bias have been to use an arbitrary weighting α which essentially requires manual tuning. It was acknowledged, however, that F-measure was also used due to the fact that the true negatives in the system could not be properly classified and was a simpler approach compared to attempting to classify type II errors in a scene (Perez & Olague 2008).

Other work on the analysis of existing evaluation metrics claims that the F-measure is more biased towards positive classification at the cost of precision and that weighted relative accuracy, which shares a connection with ROC, is a more unbiased version of accuracy, albeit “However, [weighted relative accuracy] is only slightly skew-insensitive a fully skew-insensitive version is tprfpr.” (Flach 2003). Compared to ROC, which scales using the real positives and predicted negatives (albeit in some cases needing to be predicted), precision, recall and F-measure are scaled by the predicted positives only.

2.10.2 Informedness and Markedness

Due to empirical analysis refuting the F-measure as a sound statistical metric (Entwisle & Powers 1998, Powers 2011, Powers 2015*b*, Powers 2015*a*), informedness has been proposed as an alternative (Powers 2012*a*). Although informedness and its definition both as a theoretical probability and theoretical statistic are recent, the simple dichotomous form of the statistic, known as Youden's J (Youden 1950), Delta P (Perruchet & Peereman 2004), or unskewered relative accuracy (Flach 2003), goes back to at least the 1880s (Peirce 1884).

According to Powers (2011), informedness and markedness is defined as follows:

"Informedness quantifies how informed a predictor is for the specified condition, and specifies the probability that a prediction is informed in relation to the condition (versus chance)

...

Markedness quantifies how marked a condition is for the specified predictor, and specifies the probability that a condition is marked by the predictor (versus chance)."

Informedness and markedness can be considered extensions of recall and precision respectively. Within the existing contingency matrix, informedness and markedness are represented thus:

$$\begin{aligned}\text{Informedness} &= \text{Recall}{+}\texttt{tnr} - 1\\ &= 1 - \texttt{fnr} - \texttt{fpr} \qquad (2.59)\\ &= \texttt{tpr} - \texttt{fpr}\end{aligned}$$

$$\begin{aligned}\text{Markedness} &= \text{Precision}{+}\texttt{tna} - 1\\ &= 1 - \texttt{fpa} - \texttt{fna} \qquad (2.60)\\ &= \texttt{tpa} - \texttt{fna}\end{aligned}$$

$$\text{AUC} = (\text{Informedness} + 1)/2 \qquad (2.61)$$

$$(2.62)$$

Informedness can be most succinctly represented on a ROC graph as the vertical distance between a given ROC data point, and the chance line. In figure 2.7, this is represented as I.

Both of these metrics, as their terms suggest, form a single metric representation of how informed a classifier is, and how marked the effect is. Informedness can be simplified as the `tpr-fpr`, which are the same metrics used in mapping ROC curves. However, unlike the pointwise version of AUC (which can also be derived from the informedness measure as shown in equation 2.61), informedness can be used to choose an optimal operating point based on the `tpr` and `fpr` data (which is also applicable for Markedness) to make specific comparisions at each available threshold interval without taking into account the entire set of available thresholds like the AUC does. Unlike ROC, which uses a [0 1] range, informedness is measured as the the performance of the classifier compared to the chance line. In ROC curves, it is the diagonal from [0 0] to [1 1], but in the case of informedness, now represents 0. Informedness has been shown to be a much better form of measuring the objective optimisation of classifiers. Other existing accuracy-based metrics can “give up” or “surrender early” as a result of not being able to integrate chance corrected evaluation during training (Powers 2013). Compared to ROC (2.10.1.1), informedness also eliminates the concerns of requiring the `fpr` to be fixed, and reduces complexity by not having to consider variance beyond a single dimension, which makes it far simpler to determine variance at each threshold.

However, the utilisation of informedness as a fitness measure for GP, or as a form of analysis in CV, is currently unexplored. Informedness has been applied in the field of GA for EEG feature selection (Atyabi, Luerssen, Fitzgibbon & Powers 2012), and informedness as a fitness metric converges on better solutions much faster than the F-measure (Pereira, Vega, Moreno, Dormido-Canto, Rattá, Pavón & Contributors 2015). However, with the previous exceptions, informedness has not been attempted. There are no examples in the literature of anyone using informedness for testing, or even comparison, even though there are researchers who are aware of it (Herrmann et al. 2014, Liang et al. 2015).

There are probably a few reasons why informedness has not been used for testing or comparison. Firstly, that classification of false negatives is either difficult, time consuming, or unpractical in situations where there is an insufficient dataset and/or testing methodology that can properly classify features. Secondly, there is a lack of awareness of informedness as a candidate, most research on the subject is somewhat new, and it has seen little adoption or traction, even though it has been known for well over a century (Youden 1950, Peirce 1884) due to the fact that situations where it can be properly applied have not been possible in many instances.

2.11 Uninvestigated Areas of Evaluation and Classification in GP/CV

It should be noted that in the majority of cases, CV and GP base much of their evaluation on recall/precision or true positives/true negatives of points detected, and largely neglect the inclusion of false negatives to account for type II errors during evaluation. The exception being some older examples of machine learning research like the Viola Jones algorithm and Agarwal (Viola & Jones 2004, Agarwal et al. 2004), and work performed by Perez which integrates the precision/recall with the F-measure (Perez & Olague 2008, Perez & Olague 2009*b*, Perez & Olague 2009*a*) to measure an equalisation between the two metrics. Though there are some cases where research attempts to classify true negatives, it has lacked a sufficient ground truth to properly classify them and are more based on assumption or approximation (Agarwal et al. 2004). Work from the last several years has not really explored metrics that include true negatives, nor made a concerted effort to include true negatives such as those employed by ROC or AUC. When it comes to keypoint performance especially, the work by Mikolajczyk and Schmid (Mikolajczyk & Schmid 2005, Mikolajczyk et al. 2005) is still considered the defacto standard for evaluating performance of detected features (Lindeberg 2013, Awad & Hassaballah 2016). This is not surprising given that the format of the data (2D image data) being analysed has also not changed. It is also unsurprising given that datasets still consist of images in most cases, yet researchers are increasingly trying to incorporate 3D data into these evaluations (as discussed in Section 2.2.5). Unfortunately, the boundary between 2D data, and 3D classifiers sees little overlap, except in a few rare cases (Yi et al. 2016), and instead 2D detectors use 2D data, while 3D classifiers use 3D data.

In sections 2.9 and 2.10.1, we introduced and discussed the types of classification and evaluation strategies shared by the computer vision and genetic programming fields over the last two decades. Evaluation strategies are wide and varied; however, with classification of keypoints, most work in the last two decades has concentrated on the work of Schmid and Mikolajczyk. In the case of the F-measure, evidence is growing that it is not a good candidate for evaluating features, and though ROC has advantages, it can be a difficult fit in CV. Alternatives that show promise (informedness) have experienced success in the areas of information retrieval and machine learning, but have yet to be properly utilised in CV or GP, even comparatively.

This state of affairs opens up a number of areas for research, especially in the sense that a virtualised ground truth can help enable new forms of analysis that existing methodologies currently lack, due to their inability to adequately quantify feature classification.

2.12 Virtual Ground Truth

3D simulation and virtualisation have branched out to a multitude of fields and potential applications that can only be hinted at here. Some more recent examples, however, include human training (Cohen, Lohani, Manjila, Natsupakpong, Brown & Cavusoglu 2013, Williams-Bell, Kapralos, Hogue, Murphy & Weckman 2015, Potkonjak, Gardner, Callaghan, Mattila, Guetl, Petrović & Jovanović 2016, Jang, Vitale, Jyung & Black 2017, Kim, Park, Yu, Kim, Kang & Choi 2017, Harris, Montero, Grant, Morton, Llop & Lin 2017, Aussedat, Venail, Nguyen, Lescanne, Marx & Bakhos 2017, Milne, Raghavendra, Leibbrandt & Powers 2018) the preservation and research of fragile research subjects (Fernández-Palacios, Morabito & Remondino 2017, Dominy, Mills, Yakacki, Roscoe & Carpenter 2018) investigative research that is too dangerous or time consuming (Begley 2017, Sportillo, Paljic, Boukhris, Fuchs, Ojeda & Roussarie 2017, Reyes & Chen 2017), or in some cases, insuffucient without proper 3D representation (Xue, Pal, Ye, Lenters, Huang & Chu 2017), solving real-world problems (Posada, Toro, Barandiaran, Oyarzun, Stricker, De Amicis, Pinto, Eisert, Döllner & Vallarino 2015, Lee & Park 2017) and including entertainment (Rietzler, Plaumann, Kränzle, Erath, Stahl & Rukzio 2017). 3D virtualisation is becoming increasingly more sophisticated, and, in turn, beneficial as a learning/assistance tool. It has become increasingly beneficial in human learning, and also in elucidating information about a scene that is difficult from imagery alone, or is impossible due to inaccesibility to the real world equivalent. Conversely, recreation of a virtual 3D object from single 2D images also shows promise in helping to bridge the gap between 2D and 3D environments (Rezende, Eslami, Mohamed, Battaglia, Jaderberg & Heess 2016) as with estimated point clouds (Fan, Su & Guibas 2017), and also with the assistance of keypoints (Wu, Xue, Lim, Tian, Tenenbaum, Torralba & Freeman 2016). The most recent work in cutting edge 3D model recreation addresses situationally optimised keypoint detectors, and could also benefit from the optimisation of those keypoints with 3D points. The establishment of a better foundation for feature classification and evaluation via a virtualised ground truth is a growing area of investigation, but currently not a great deal has been done to extend 2D feature evaluation into 3D ground truths to enable better feature classification and evaluation.

2.12.1 Thesis Motivations

Based on the review of literature in the field of CV and GP, there seem to be a number of unanswered questions that this dissertation hopes to investigate, and address.

In summary, regarding the existing CV field, it can be identified that:

- Schmid and Mikolajczyk's work on keypoint/image descriptor repeatability is highly constrained to 2D-based datasets that require calibration to establish ground truth, which greatly limits testing flexibility.
- Schmid and Mikolajczyk's work on keypoint/image descriptor repeatability uses an error threshold (more commonly referred to in this dissertation as $\epsilon = 1.5$, also known as the Moore neighbourhood, to classify two points as "repeated" or not, with no established evidence to explain why $\epsilon = 1.5$ is the appropriate choice.
- In addition, a homography is used exclusively to approximate the 3D positions of points, which is applied to a 2D pane of points, as such, it cannot be applied to dimensions beyond 2D to properly identify foregound/background features. Therefore, point clouds cannot adopt this methodology as it stands.

The work by Schmid and Mikolajczyk is still regularly used in the field of CV, and notably, in GP (Shao et al. 2014) when it comes to keypoint evaluation and performance testing.

For GP, we identify that in general, precision, recall and the F-measure are heavily relied-upon to evaluate classified features in CV contexts. ROC analysis has been largely overlooked in recent years, as it requires true negative classifications, which are very difficult to ascertain. The other evaluation methodologies exclude true negatives due to more constrained methodological testing. Less comprehensive alternatives are used under the justifications of being better alternatives like F-measure, (which are essentially a compromise), due to being unable to be properly determine the true negative classifications. The failure to embrace better alternatives seems to be in part due to a poor awareness of more recent resurgence in alternatives like informedness, or limitations in the types of ground truth that are used, namely 2D-marked up datasets of images.

Most of the issues identified here stem from the use of 2D images of 3D scenes. The focus on 2D images as a ground truth in CV and GP seems straightforward conceptually due to that being the only form of ground truth available, but as the use of a homography suggests, there is a stronger need to have more information about features than simply the 2D image data. The field of CV is limited by what it can do with 2D datasets alone and though there is a concerted effort to incorporate 3D and other applications, such as colour, or more complex image descriptors, scene depth is undeniably becoming a valuable and in more and more cases, necessary component for analysis of classifiers.

The original intention of the use of a homography is to emulate the positions of features based on a different viewpoint (thus emulating depth), which insinuates that a sufficient ground truth requires the depth of the features within a scene if adequate classification is to take place. Otherwise, as in the case of Agarwal et al.'s research (2004), the computation complexity in feature classification increases dramatically and greatly limits the ability to optimise feature classifiers. The computational complexity can only be adequately reduced if depth plays a key role in the feature classification process. As Agarwal et al.'s later work demonstrates (Agarwal et al. 2009). In the case of more recent work in the CV/GP field, there is still a heavy reliance on 2D datasets and a homography to classify real-world 3D features, and though there has been a move towards image-descriptor centric optimisation strategies, the dependence on 2D marked-up datasets, a homography, and Schmid-based repeatability metrics (which has not changed for keypoint repeatability since 2000) remains unchanged. The use of a homography is rather counterintuitive for GP applications, as the classifiers are trained using an approximate ground truth during training but then using an optimisation based on an approximation for real-world feature positions. Of particular note, even the bleeding edge of current research employing GP utilises image sets (Shao et al. 2014, Al-Sahaf et al. 2017, Rodriguez-Coayahuitl, Morales-Reyes & Escalante 2018, Bi, Zhang & Xue 2018) and does not utilise a depth metric when measuring fitness of features.

The limitations of CV and GP, as previously stated seem to be intrinsically inter-related where one depends on the limitations of the other, and thus retards the adoption of better alternatives. Specifically, the Schmid based repeatability for 2D detectors can't fully test performance of 2D classifiers with 3D scenes, and existing 3D classifiers are not isolated sufficiently that their 2D performance can be ascertained, due to a dependence on 3D scene data. As has already been noted in this literature review, there are cases where research in the field of GP has acknowledged that empirically better alternatives like the informedness measure are available (Herrmann et al. 2014, Liang et al. 2015), but still avoids using them, even as a form of comparison.

As outlined in chapter 1, this thesis will utilise the benefits of a virtualised 3D environment to emulate the scene's ground truth, such that more precise classification and evaluation can be performed.

Chapter 3

A Virtual 3D Ground Truth for Repeatability

3.1 Introduction

Based on the existing literature in the field (covered in Chapter 2) there are a number of areas in the field of CV and GP that have not been explored, or that are open for future investigation. As a foundation, one of the most vital elements required to test these investigations, is a framework that can utilise existing interest point detectors, and virtual 3D spaces. At a low level, the points that are detected by any keypoint detector, or image descriptor detector should be testable using existing methodologies.[1]

3.1.1 Chapter Goals

As covered in Chapter 2, a vital component of measuring the performance of interest point detectors is the repeatability of points. In this Chapter, the main focus is to test the accuracy of repeated points in a virtual 3D space in a manner that is not only accurate, but fast, and flexible. Ideally, such a method will allow the automation of tests to be performed on large and complex datasets which can

[1] These experiments were published in the following proceedings/publications, and are reprinted with permission from Copyright ©IEEE 2013 (Lang, Luerssen & Powers 2013*a*), Copyright ©Springer International Publishing Switzerland 2013 (Lang, Luerssen & Powers 2013*b*), and Copyright ©IGI Global 2014, by invitation, Int. J. Softw. Innov. (Lang, Luerssen & Powers 2014). Black and white figures are presented in this chapter as is, in their published form.

leverage the spatial depth in a scene, in order to properly represent the location of an interest point. This information, which a conventional interest point detector cannot use when processing images, can be exploited during the training process, as demonstrated by Olague and Trujilo's work (2011), to more accurately discern interest point positions in a scene, as well as to help discard false positives. This more accurate performance measure can then be used during the training process to better select for candidate detectors without interfering with the basic structure of traditional interest detectors, as well as not requiring overly complex real-world testing schemes. All the performance testing of a complex scene can be performed in a virtual 3D space with an extremely high degree of accuracy and flexibility via readily-available affine transform tools.

With this goal in mind, Chapter 3 will test and compare nine conventional interest point detectors using evaluation and testing methods based on Schmid (2000). Each detector is tested on the BSDS500 image set using rotation in the X, Y, and Z axis as well as scale in the X,Y axis. These results demonstrate the differing performance and behaviour of each detector across the evaluated transformations, which will help enable better discrimination between candidate detectors during the training process.

To summarise, the following research objectives will be carried out.

- Create a virtualised scene that can be properly integrated with standard 2D-based keypoint detectors and image descriptors.
- Formulate a set of scene conditions (mainly affine transforms) that can be applied to the virtual scene, and which satisfy most testing conditions as decribed in Section 2.2.1.
- Duplicate the conventional repeatability described in the literature review (Section 2.9.1), which can be utilised for 3D virtual scenes.
- Demonstrate that 2D detectors perform reliably in virtualised scenes regardless of whether 2D or 3D datasets are utilised. In essence, demonstrate that 2D detectors can emulate the use of a homography of 2D points for 2D datapoints, as well as utilise interest points projected into 3D space.

3.1.2 Chapter Organisation

The rest of this Chapter will be split into 2 major sections. While both sections will cover all the above points, each section is split into 2D and 3D respectively.

The establishment of the first section's testing serves as the foundation for the more complex investigation into the performance of interest points that need to be manipulated in 3D, with a dataset of 3D-scanned, and human-made models of differing varieties. Finally, a summary of the investigations are made.

3.2 3D Ground Truth

3.2.1 System for Testing Evolved Interest Point Repeatability

The *System for Testing Evolved Interest Point Repeatability* (STEIPR) was created to address the measuring of interest point repeatability, with the intent of evolving interest point detector candidates, hereafter referred to as STEIPR. The system was designed to be fast, flexible, and focused primarily on measuring interest points in a 3D scene. STEIPR, at its core, had to be robust in order to function as a testing suite that could handle many points quickly and accurately. To ensure that each of these goals could be met, STEIPR was completely written in C++ so that it remained compatible with 3D tools like OpenGL, as well as to make the codebase as efficient and fast as possible, as it was intended to test thousands of candidate detectors, with potentially millions of interest points. Unlike conventional 2D scenes where a homography can be easily calculated, a large focus was on exploiting OpenGL's affine transform tools, which allowed fast and efficient manipulations of points and the models that are displayed within these virtual spaces. The virtual space also had the added benefit that 2D scenes could also be rendered and compared with no modification to the STEIPR system by rendering a flat pane with the image as a texture. To test repeatability, the STEIPR system rendered a candidate model, and followed these steps in sequence:

1. *Render scene I_1 (reference scene) with selected affine transform and model.*
    - *1.1 Run detection on 2D image of rendered scene.*
    - *1.2 Convert 2D interest points to 3D world co-ordinates.*
    - *1.3 Remove 3D points not within model dimensions.*
    - *1.4 Optionally order by point strength and trim to max points.*
    - *1.5 Apply inverse affine transform to 3D points.*

1.6 Convert transformed 3D points to 2D points ($\widetilde{x}_1$).

2. Repeat 1.1-1.6 for scene I_i.

3. Measure Repeatability

3.1 Find closest $\widetilde{x}_i$ in I_i, to each $\widetilde{x}_1$ in I_1.

3.2 Measure distance of point pairs.

3.3 Calculate repeatability measure.

With this process, Schmid repeatability can be calculated with a high degree of accuracy. This helps to filter out false positive points that may not be a part of the model itself with the help of a model bounding box, like the background borderline around the model, and also reduces the chances od occluded points. Another benefit of this approach is the fact that in a 2D scene the scene depth can't be discerned, such that what may look like two neighbouring points on a model can't be discriminated. However, in a 3D situation, the distance of a 3D feature based on a 2D point can be accurately quantified and this prevents distorting of the repeatability measures. As such, all tests have utilised the 3D distance of neighbouring points in order to exploit this new information. A slight downside to this approach, however, is the fact that finding the distance in this manner is computationally expensive. Nevertheless, the expectation is that computational costs can be at least somewhat minimised by only processing a maximum number of points during training, though in this chapter, the detectors have not been artificially restricted.

As the STEIPR processing sequence shows, the conversion from 2D, to 3D, and back to 2D, allows us to take advantage of the 3D space without creating extra requirements on the detector being tested. This saves the detector from having to store any extra information or know anything more about the scene than it normally would. It also allows the detector to seamlessly integrate existing conventional detectors and enable comparisons based on the same ground truths and datasets when testing. In essence, the only information passed to the candidate detector being tested is an image from the camera viewport of the scene, and the only information received is the x,y positions of each point found, and each point's strength when applicable. This consistency across detectors enables STEIPR to measure performance in a consistent manner.

3.3 Experiment I

3.3.1 Methodology

As covered in Chapter 2, the most common method of evaluation of interest points in a scene is based on repeatability. By measuring the repeatability of interest points in two slightly different scenes I_1 and I_i, we can determine how well a given detector performs (Schmid et al. 2000) and how well the detector can track these points across the applied transformation. Consider that we compare detected interest points x_1 and x_i from I_1 and I_i. The distance between any two given interest points can be determined using the homography H_{1i} of an interest point in two different images of a scene, as shown in figures 3.1 and 3.2. The repeatability of interest points is gauged by their locality within a radial threshold ϵ. Any interest points that appear in both scenes (denoted $\widetilde{x}_1$ and $\widetilde{x}_i$) and are within the threshold distance are considered repeated. This determines a set of "repeated" points as:

$$R_i(\epsilon) = \{(\widetilde{x}_1, \widetilde{x}_i) | dist(H_{1i}, \widetilde{x}_1, \widetilde{x}_i) < \epsilon\} \tag{3.1}$$

A repeatability rate from 0 to 1.0 is derived from this as a ratio of repeated features divided by the lowest number of detected features between the image pairs. This is represented as:

$$r_i(\epsilon) = \frac{|R_i(\epsilon)|}{min(n_1, n_i)} \tag{3.2}$$

where $n_1 = \{|\widetilde{x}_1|\}$ and $n_i = \{|\widetilde{x}_i|\}$. However the convention of curly braces to represent a set in Schmid's work is not the standard convention in this thesis and it should be noted that it is omitted in preference of upper and lower case for sets and elements of sets. The use of repeatability measures has led to numerous optimisations of existing detectors, including optimisations of affine and scale invariance (Mikolajczyk & Schmid 2002), finding detectors that complement their strengths and minimise weaknesses (Guillaume Gals 2010), and the evolutionary design of interest point detectors (G. & L. 2011). Repeatability has also assisted in the reconstruction of 3D meshes based on the interest point clusters generated (Sipiran & Bustos 2010), and seen real-world applications in which interest point features are tracked so that camera motion, position and surface distance can be determined (Hartley & Zisserman 2003).

When it comes to handling repeatability in 3D, a slightly different approach

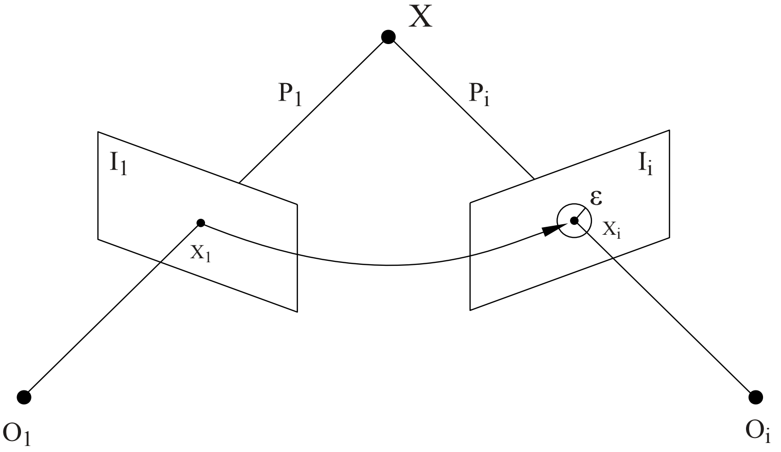


Figure 3.1: Schmid repeatability: the points x_1 and x_i are the projections of a 3D point X onto images I_1 and I_i. A detected point x_1 is repeated if x_i is detected in the ϵ-neighbourhood of x_1.

is used. When the interest points x_1 are detected in the image of a scene, I_1, the scene is then changed by an affine transformation T_{1i} into a different scene and corresponding image, I_i, that has a different set of interest points x_i. Unlike Schmid et al., we operate on virtual 3D spaces, so T_{1i} as well as the projection transformations P_1 and P_i (described as projection matrices by Schmid (2000)) are precisely known. This allows a ray to be cast into the scene for each point in x_1 to determine, after intersection with the 3D model, which particular 3D point X (which is described as a singular 3D point by Schmid (2000)) in the scene corresponds to it; these points are subsequently also transformed by T_{1i} and projected into $\widetilde{x}_1$. The repeatability of each interest point in x_1 can then be gauged by whether there is an interest point in x_i around each transformed $\widetilde{x}_1$ within a radial pixel distance threshold ϵ. The process is illustrated in figures 3.1 and 3.2 and can also be applied to x_i via an inverse T_{1i}^{-1}. Interest points must be located within the viewport after transformation. I.e., if points from $\widetilde{x}_1$ appear in I_i and $\widetilde{x}_i$ points appear in I_1 then those points are considered to share the same working area and are included for testing of repeatability. Use of a 3D virtual space allows us to manipulate interest points easily so that they can be measured, regardless of whether the object is a pane or model, although we shall focus on 2D image panes in this chapter. The 3D scene S_i containing the image pane is loaded into an OpenGL rendering context and transformed by matrix T_i, then orthographically projected into image I_i, so that a given interest point detector can locate a set of 2D points X_i on it. We then inverse project from each point x, y in X_i into the scene to determine the corresponding part of the scene that is producing this feature of interest. This produces a set of 3D interest points, C_i.

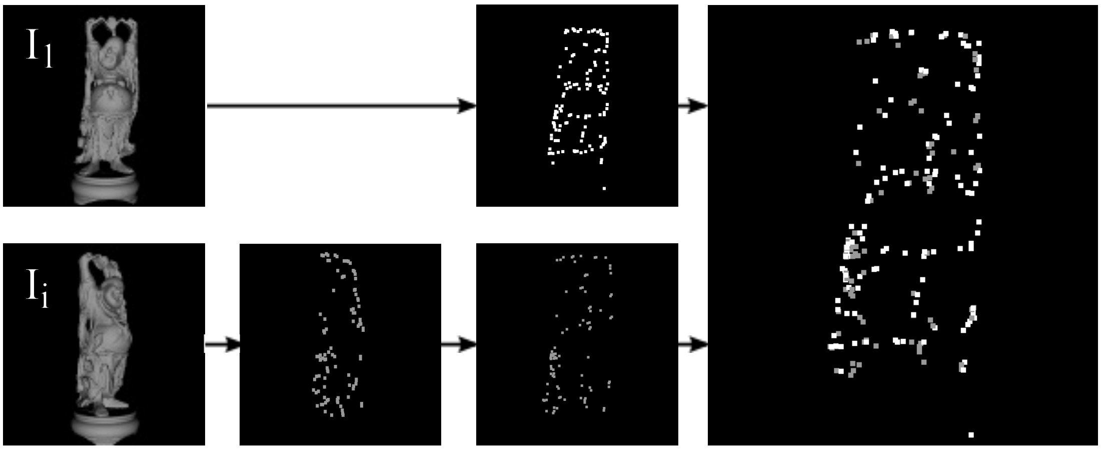


Figure 3.2: Illustration of 3D model being rotated from 0° shown in I_1, to 50° in I_i. I_1 and I_i are processed to produce interest points x_1 and x_i. 3D correspondences are determined via inverse projection and rotated by 50°, so that each point in x_i can be mapped to I_1 (and vice versa, not shown). Repeatability can now be measured easily by testing for points in close neighbourhood.

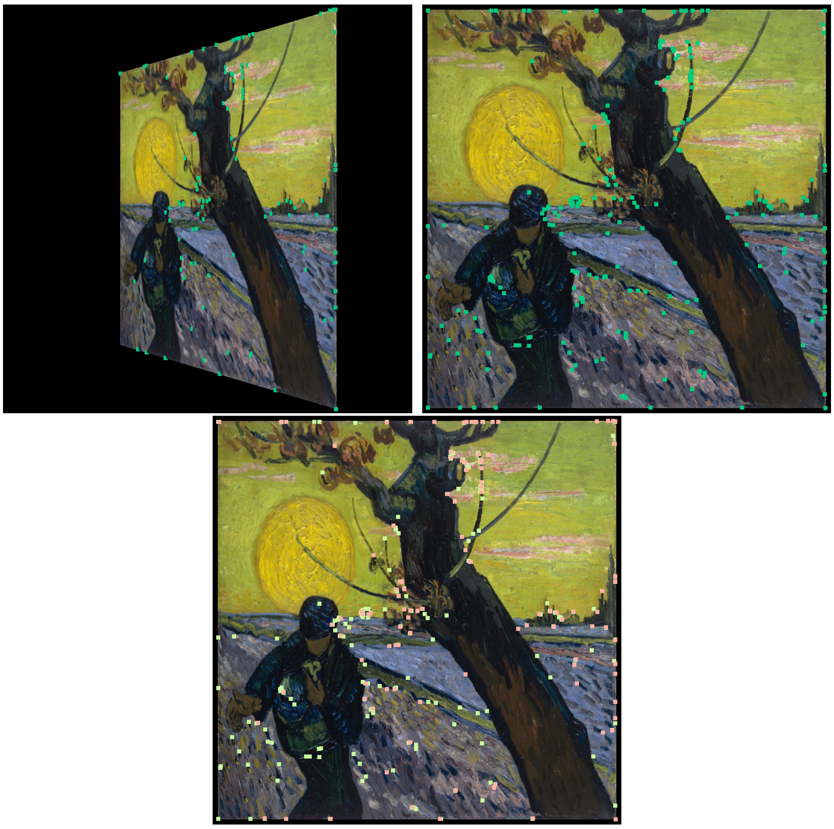


Figure 3.3: Y axis rotation with Foerstner detection of Sower image, with first image at -50°, 0° and both overlaid using matrix transforming of first image's interest points.

Depending on the content of the scene and the nature of transformations, trimming of C_i may be necessary. In situations such as ours, where a single plane within an otherwise empty scene is used, interest points that detect edges can fall outside its boundary, subsequently not residing on the plane's surface but on the far clipping plane of the view frustrum. These are removed using a bounding box around the model(s) in each S_i. Each point's x, y, z position in C_i is compared to the dimensions of the bounding box and culled if it does not lie within. From this culling, a set of trimmed points $\widetilde{C}_i$ is created.

$\widetilde{C}_1$ represents the reference interest points of the scene without any transformation applied, whereas every other $\widetilde{C}_i$ is the result of a particular scene transformation T_i. To measure repeatability, the inverse of the scene transformation T_i^{-1} is applied to $\widetilde{C}_i$, so that these points share the same transform state as the reference detection $\widetilde{C}_1$. The resulting "untransformed" points $t_i\widehat{X}_i$ are then projected back into a 2D image and compared to trimmed set $\widetilde{X}_1$. Schmid's measure of repeated points can now be simplified to:

$$R_i(\epsilon) = \{(\widetilde{x}_1, \widehat{x}_i) | dist(\widetilde{x}_1, \widehat{x}_i) < \epsilon\} \tag{3.3}$$

In order to reproduce Schmid's original experiment (Schmid et al. 2000), the Sower painting is used as an image to manipulate within the 3D space. Additionally and separately, automated testing is also performed on 500 images from the BSDS500 image set (Martin et al. 2001). The viewport resolution is 800x800 for the sower image and 400x400 for the BSDS500 set. The image's original aspect ratio has been ignored and a uniform aspect ratio has been used to better facilitate automated testing. The images are fitted on a square pane to ensure the image is fully centered, and positioned in 3D space so that it fills the viewport but does not occlude at any stage. To demonstrate the utility of the 3D environment, as well as to emulate Schmid's testing methods, simple affine transforms are used to rotate and scale the pane. This encompasses rotation in the X,Y and Z axis, as well as uniform scaling in the X,Y axis, relative to the position of the viewport. The detectors being tested are Harris (Harris & Stephens 1988), KLT (Kanade April 1991), Fast (Rosten & Drummond 2006, E. Rosten 2006), SIFT (Lowe 2004) and SURF (Herbert Bay 2006). Vigra (Käthe 2000), a computer vision library, is also included in the evaluation. It implements the Harris (Harris & Stephens 1988), Rohr (Rohr 1992), Foerstner (Förstner 1986) and Beaudet (Beaudet 1978) detectors. The detectors' settings are left at default values. Each test used linear and anisotropic filtering when taking texture scaling into

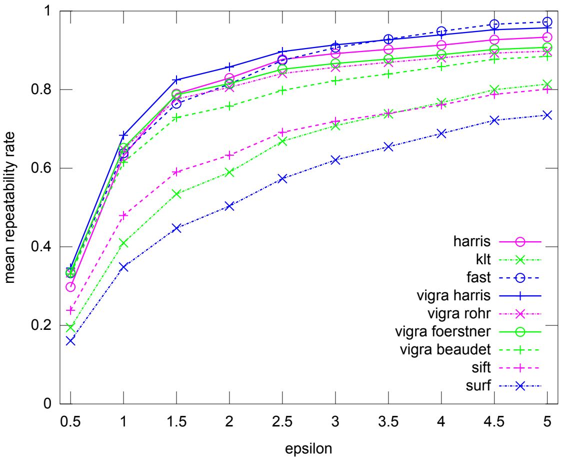


Figure 3.4: BSDS500 set. Mean ϵ for all tests.

consideration. The Van Gogh Sower image was sourced from `http://0.tqn.com/d/arthistory/1/0/6/m/vvg_cotn_moma_10.jpg`[2]. Each test uses a reference image to compare with each detection. Each detection is not compared linearly in a back-to-back manner. Instead, a reference image is nominated and all other detections compared to the reference to determine repeatability.

3.3.2 Results and Discussion

Firstly, it is worth noting that Schmid (2000) set ϵ=1.5, because it was a good balance between strong repeatability and had a low likelihood of including nearby unrelated interest points on the Sower painting. This is further supported within our 3D testing environment which used the larger BSDS500 set, as illustrated in Figure 3.4, and ϵ=1.5 will therefore continue to be used in subsequent comparisons.

The increments used for all of our tests were similar but not identical to those in Schmid et al.'s work (2000), where the rotation positions and scalings weren't fully documented. Rotation in the Z axis was performed from 0° to 180° in 10° increments, with the reference at 0°. Rotation in the Y axis began at -50° and extended to +50°, in increments of 5°, with the reference at 0°. Scale was from factor 1.0 (no scaling) to 4.0. With the reference at 1.0, the image was scaled in 0.25 increments. X rotation, which had not been carried out by Schmid, was tested for completeness and was the same as Y axis rotation. With the ability to test large image sets, it becomes possible to make general observations about which detectors are best suited for repeatable detections given certain criteria. After

[2] An updated image location can be found at `https://commons.wikimedia.org/wiki/File:Vincent_van_Gogh_-_The_sower_-_Google_Art_Project.jpg`

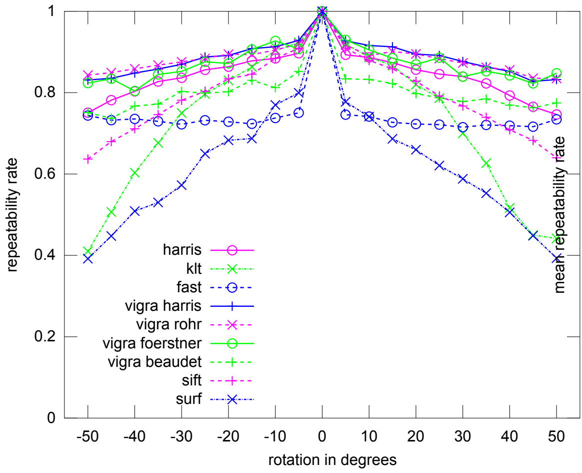


Figure 3.5: Sower image. X axis at ϵ=1.5 .

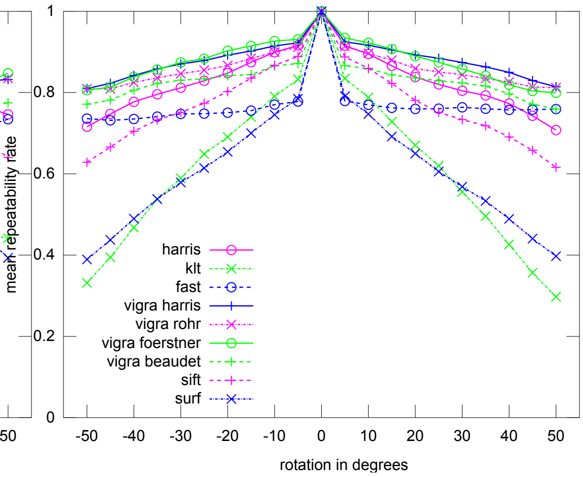


Figure 3.6: BSDS500 set. X axis at ϵ=1.5 .

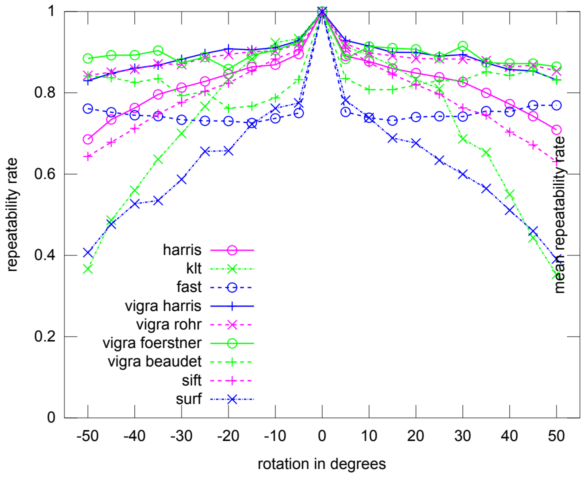


Figure 3.7: Sower image. Y axis at ϵ=1.5 .

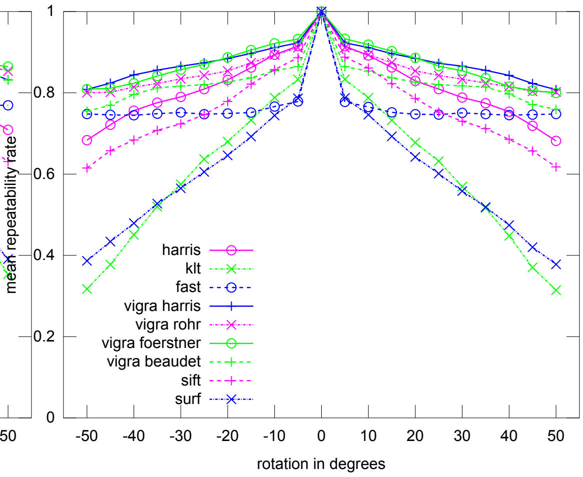


Figure 3.8: BSDS500 set. Y axis at ϵ=1.5 .

testing each detector on the BSDS images, we could see that SIFT and SURF were particularly susceptible to affine rotations greater than 30° (Lowe 2004), and this was reflected in the results for Y and Z rotation in figures 3.8 and 3.10. All other detectors showed a high degree of sensitivity to orthogonal angles in Z rotation repeatability. Y rotation in figure 3.8 showed similar responses with each detector. The FAST detector's Y rotation result deviated from other detectors by returning a very stable, if somewhat lower, repeatability rate even when the angle of rotation became extreme. Schmid et al. noted in their tests that camera noise contributed to results where Z and Y rotation at 0° was less than 1.0 (Schmid et al. 2000). In the results presented here, camera noise was completely absent, since a virtual

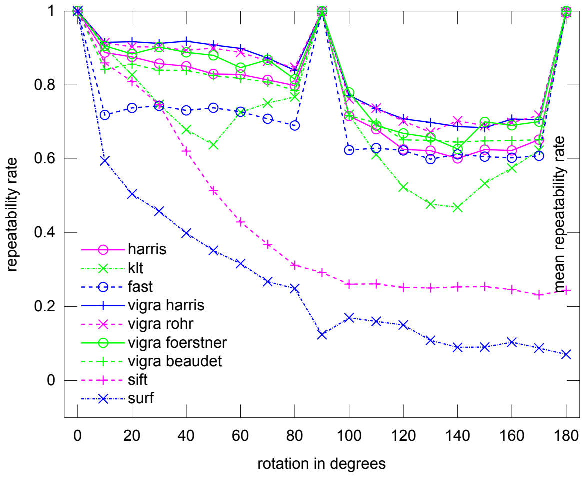


Figure 3.9: Sower image. Z axis at ϵ=1.5 .

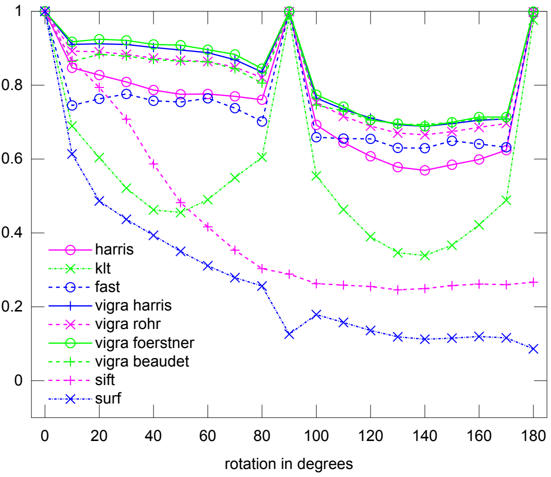


Figure 3.10: BSDS500 set. Z axis at ϵ=1.5 .

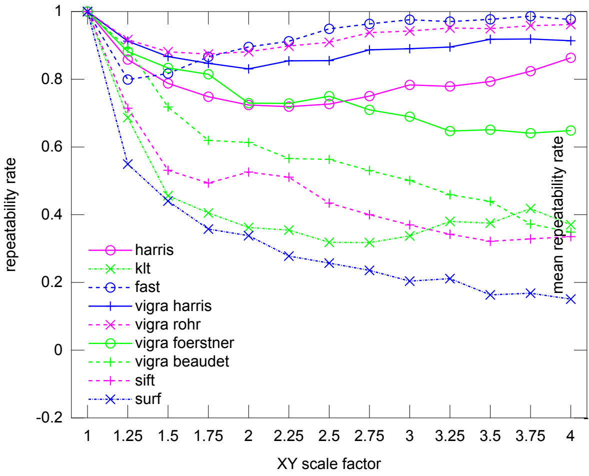


Figure 3.11: Sower image. XY scale at ϵ=1.5 .

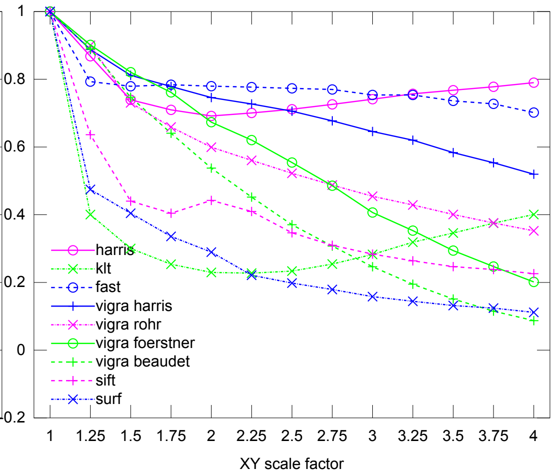


Figure 3.12: BSDS500 set. XY scale at ϵ=1.5 .

environment can avoid noise traditionally inherent in manual tests. This resulted in repeatability of 1.0 every time when comparing the same scene to itself. Further investigation into the impact of various filtering methods on detection results will clearly be necessary to determine the best options for reproducing real-world results.

Contrasting the above with the original Sower image in figures 3.5, 3.7, 3.9, and 3.11, we can better determine the general effectiveness of each interest point detector. There were marked differences in detector performance, most notably in scale. Scale XY in figure 3.12 showed a varied response. Beaudet, Foerstner and Rohr showed initially strong responses but quickly degraded as scale increased, demonstrating in this situation that the detectors' invariance to scale was probably weak. Harris, Vigra's Harris and FAST showed a fairly consistent level of repeatability as scale increased, indicating that they were more resistant to scaling effects when using this particular image set. SIFT, SURF and KLT show rapid degradation in repeatability at small scale changes, yet KLT tends to improve at larger scales.

In these tests, the number of interest points found across detectors varied greatly during testing, from a few hundred, to thousands. In the context of measuring repeatability, Schmid's approach gave no weighting as to whether a greater number of repeated points was better or worse, only a bias towards a higher proportion of repeatable points than unrepeatable points. This resulted in a bias towards detectors that had a larger number of interest points, as more of them were likely to fall within the repeatability distance threshold by simple chance. Conversely, detectors tracking very few but very salient interest points could also appear to perform extremely well, but be of limited practical use. The optimal number of interest points within a scene remains an unexplored factor when calculating repeatability in this way.

3.4 Experiment II

This experiment built on the previous work by testing the same detector. However, in this instance, the testing was conducted on 3D models. This was to demonstrate the performance of interest point detectors on 3D virtual spaces. The following sections continue to test the robustness and flexibility of the STEIPR system, by taking full advantage of the depth and detail of 3D models in a scene.

3.4.1 Methodology

When representing the position of interest points in Euclidean space, and because 3D data (ie. scene depth) is being utilised as opposed to a flat pane to represent a 2D image, the "$\hat{}$" for untransformed 3D interest points x_i, in Equation 3.3 has been removed. This is because the scene is no longer using approximations of features from 2D images, as the 3D objects themselves represent a genuine ground truth. The Schmid repeatability is now represented as:

$$R_i(\epsilon) = \{(\widetilde{x}_1, x_i) | dist(\widetilde{x}_1, x_i) < \epsilon\} \tag{3.4}$$

and represents the use of Schmid repeatability in subsequent Chapters.

The models were manipulated within an OpenGL world space and rendered in a 400x400 pixel OpenGL viewport using linear filtering for textures. The models used were a diverse combination of 34 scanned and artist-made models. Figure 3.13 shows a small subset of these: 15 were scanned and untextured, 10 were artist-made, textured models, 8 were scanned, textured models, and one was an unscanned, untextured, artist-made model. All models were sourced from various research and model repository sites [3][4][5][6][7]. Each model's initial rest position was oriented to help minimise occlusion and easily fit within the viewport. Interest points that exceeded the viewport area at any stage before or after transformation were excluded from the repeatability score. Interest points that did not correspond to any feature of the model were also culled by establishing a bounding box around the model's dimensions. These culled interest points mostly included those existing just beyond the edge of the model, which would inverse-project to the distant backplane of the 3D scene, and therefore did not practically contribute to the repeatability score.

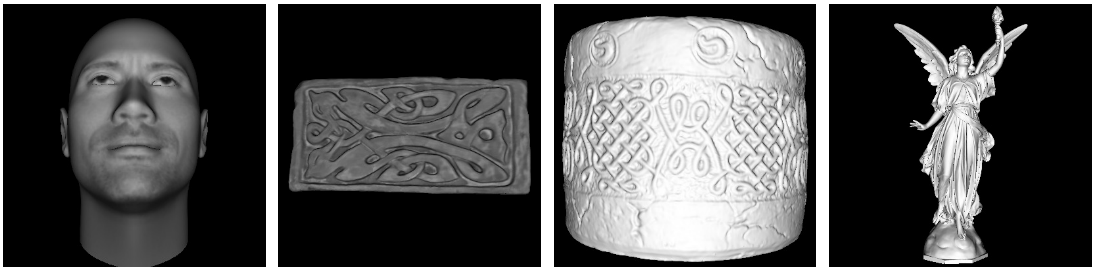


Figure 3.13: Sample of models used in testing.

The structural variability of the models necessitated that the robustness of each detector was tested on as many model positions as possible – an approach also followed by Schmid's original work. Since many applications for interest point detectors are applied to multiple, sequentially-linked frames (Hartley & Zisserman 2003), it seems reasonable to develop a testing regimen that would fit this. Schmid's approach of limiting rotation to within -50° to 50° was retained for X and Y rotation, since occlusion is much more likely at more extreme rotations, as well as rotating 180° face-on in the Z axis. XY scale (i.e., zooming into the

[3] http://users.cms.caltech.edu/˜kmcrane/Projects/ModelRepository

[4] http://people.csail.mit.edu/tmertens/textransfer/data/index.html

[5] http://graphics.stanford.edu/data/3Dscanrep/

[6] http://www.sci.utah.edu/ wald/animrep/

[7] http://www.turbosquid.com/3d

image) was arbitrarily capped to a scaling factor of 4.0. Incrementation for X and Y rotation was set at 5°, which means that the total 105° rotation was covered in 21 frames. Z rotation was made in 10° increments and ranged from 0° to 180°. The X, Y and Z rotations were all compared to the rest position at 0°. Scale in the XY axis started at 1.0 in the rest position and was incrementally increased by 0.25 up to the aforementioned maximum.

The noise filter used for testing was a simple luminance based filter that randomly modified the luminance of each pixel within a specified range (tolerance) of the total luminance space; a tolerance of 1.0 would have produced completely random luminance values along a uniform distribution. Each of the repeatability evaluations were done using no noise, noise with a tolerance of 0.05 and noise with a tolerance of 0.10. The resultant image graininess assists in determining how reliable the feature detectors were when the intensity of pixels were changed slightly and randomly, as happens in natural image capture. For a simple illustration, Figure 3.14 demonstrates how the noise filter affected the luminance of pixels in the image.

3.4.2 Results and Discussion

Interest point repeatability depends on the chosen ϵ value. Lower ϵ values are generally preferable, as they require interest points to lie very close together to be regarded as describing the same image feature, whereas larger values might lead to confusion between different features. However, the low value of 0.5ϵ, where two interest points correspond to each other only if they are closer than half a pixel a[art in distance, yielded very poor performance in our tests, as shown in Figure 3.15 for each level of noise across all tested transformations. Further investigation revealed that this could be reproduced by simply changing the direction of the scanning subwindow of the detectors, so this level of inaccuracy appears to be surprisingly intrinsic and common to the detector implementations. Results at 1.5ϵ, i.e. a single pixel tolerance within the full Moore neighbourhood, may hence be more representative for purposes of comparing these detectors, and for this reason and for the sake of brevity, we will focus on 1.5ϵ when illustrating further outcomes.

Each of the transform tests in figures 3.16, 3.17, 3.18 and 3.19 illustrate the mean repeatability of each detector for all 34 models tested. Each level of noise tested shows that the performance of most detectors degraded as more noise was introduced. In some cases, such as where the Harris detector was used, the

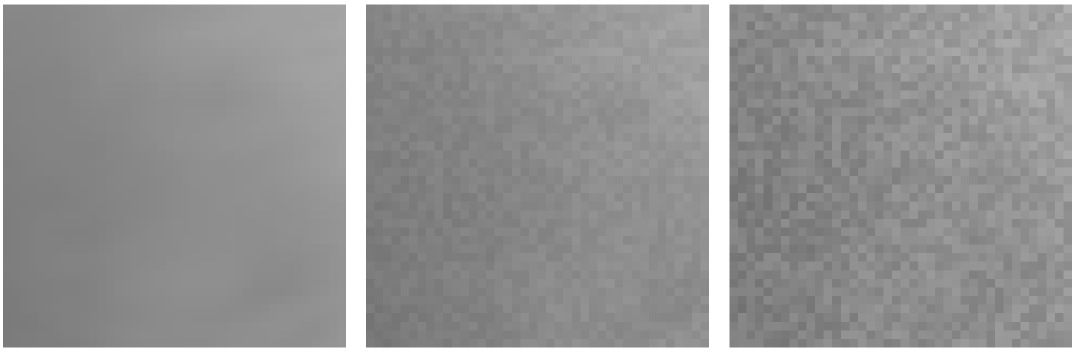
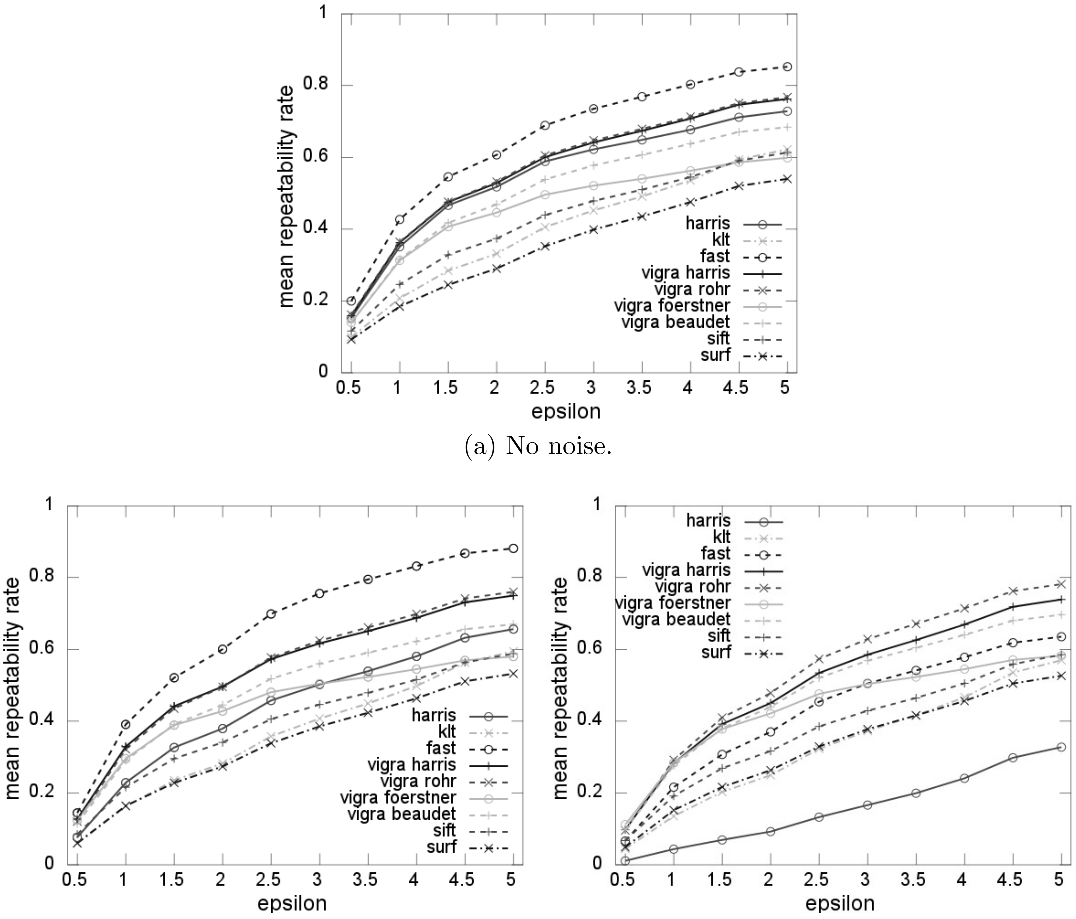


Figure 3.14: Zoomed-in sample of noise filter with no noise, noise with 0.05 tolerance and noise with 0.10 tolerance.

performance degradation was quite marked. In every test, Harris with default settings demonstrated very little robustness as more noise was introduced, which is observable even at the lower noise level, but can most easily be seen in figure 3.15c. Interestingly, the FAST detector had the best overall performance, and even performed unexpectedly better with a small amount of noise (see figure 3.15b), although it suffered a steep performance decline—particularly for small transformations—at the higher noise level tested here.

Overall, compared to the other interest point detectors, the FAST detector was most robust across larger transformations, as demonstrated in figures 6 to 9. Only at rotations of less than 10°, scale changes under 25%, and very high noise levels did the other detectors remain competitive. Conversely, SIFT, SURF, and KLT constituted the bottom of the pack, with weak capabilities in following features across larger model rotations and scaling. However, it is worth remembering that this assessment was only based on a single performance measure—repeatability—and there are other properties of these detectors that can make them desirable for a particular application. The Vigra-included detectors performed within a comparatively similar mid-range performance band, and there are few surprises to be noted here. As with all other detectors—although only weakly observed with SIFT and SURF—they exhibited a particular sensitivity to multiples of 90° rotations along the Z-axis.

Though some effort was taken to reduce the effect of point occlusion on repeatability, the models themselves may self-occlude features when rotated along the X and Y axes. Conventionally, points occluded in this way should be considered to be outside of the working area, but such situations are not fully handled by our system. At this stage, we do not know how much of a significant factor this is as far as repeatability is concerned, although large rotations would obviously result



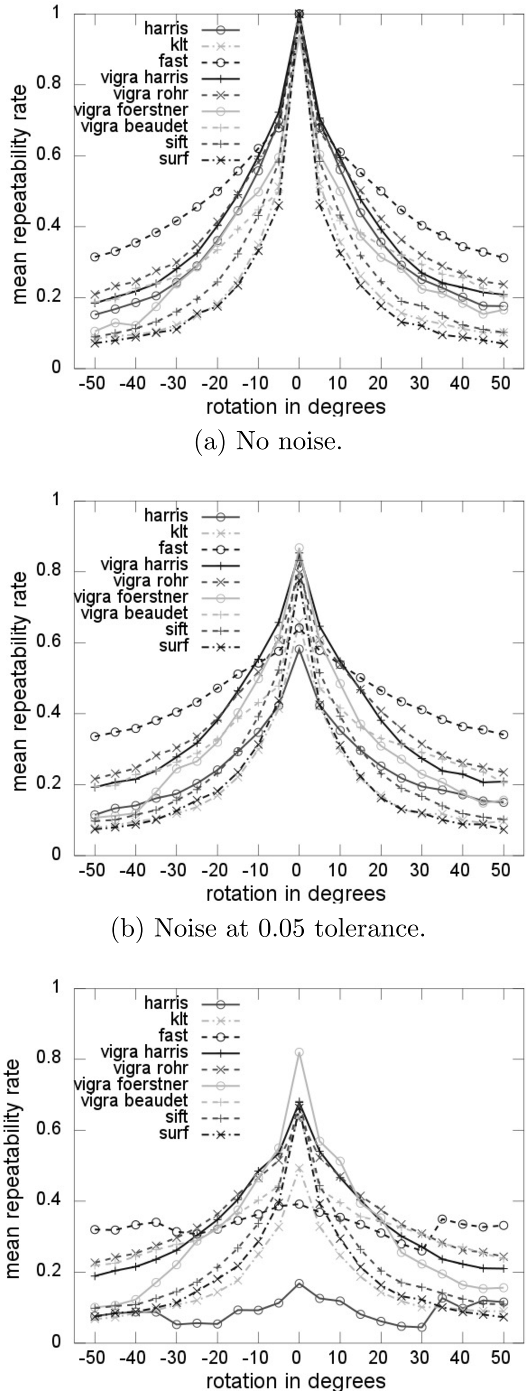
(a) No noise.

(b) Noise at 0.05 tolerance.

(c) Noise at 0.10 tolerance.

Figure 3.15: Mean of 0.5ϵ to 5.0ϵ, for all tests, with all models.

in extensive occlusion. Because depth does not play a factor in measuring the ϵ distance of points in images I_1 and I_i, we can find situations where interest points at a greater relative Z depth and closer relative to the XY axes can be considered closer than interest points that share the same Z depth yet are measured as further away from the reference interest point in image I_1. However, situations like this are only a problem if the point from I_i is also an occluded point in I_1. A larger impact on repeatability is expected from those interest points that arise along the edge of the model (bordering the scene background) but which do not represent an actual feature on the model itself. These do not consistently transform with the rest of the model, i.e., a rotation of a sphere will not change anything about the position of the sphere's edge. Currently, such interest points are not filtered out (except if they lie outside the actual model), as they are very distinct features of the image and obvious targets for image point detectors, but more careful consideration of their impact is needed.

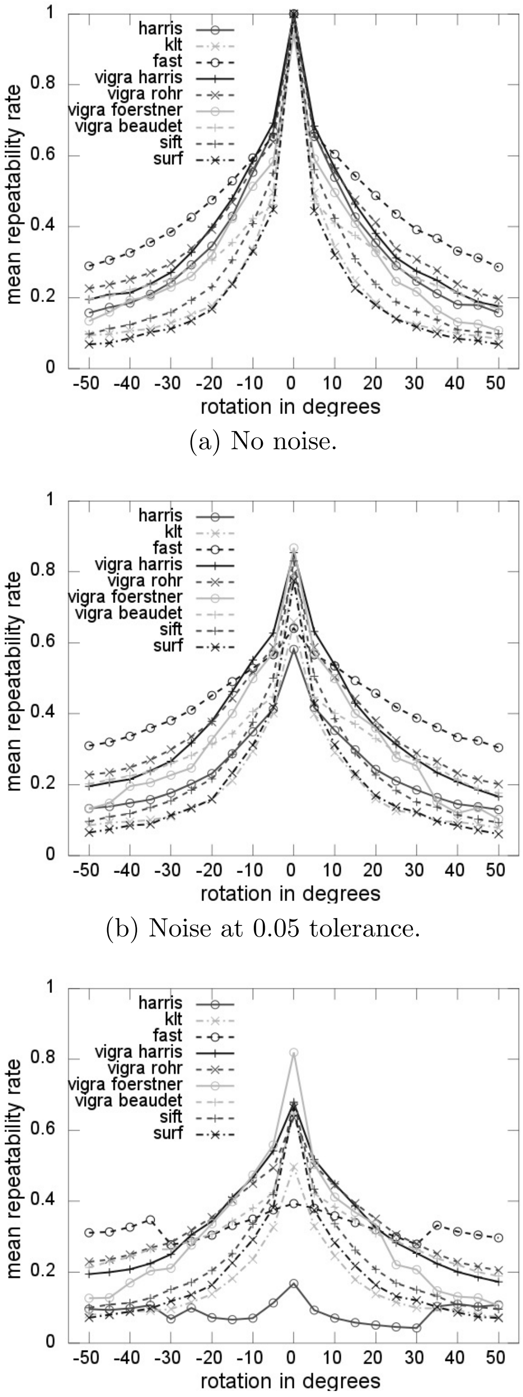


(a) No noise.

(b) Noise at 0.05 tolerance.

(c) Noise at 0.10 tolerance.

Figure 3.16: X rotation at 1.5ϵ with all models.

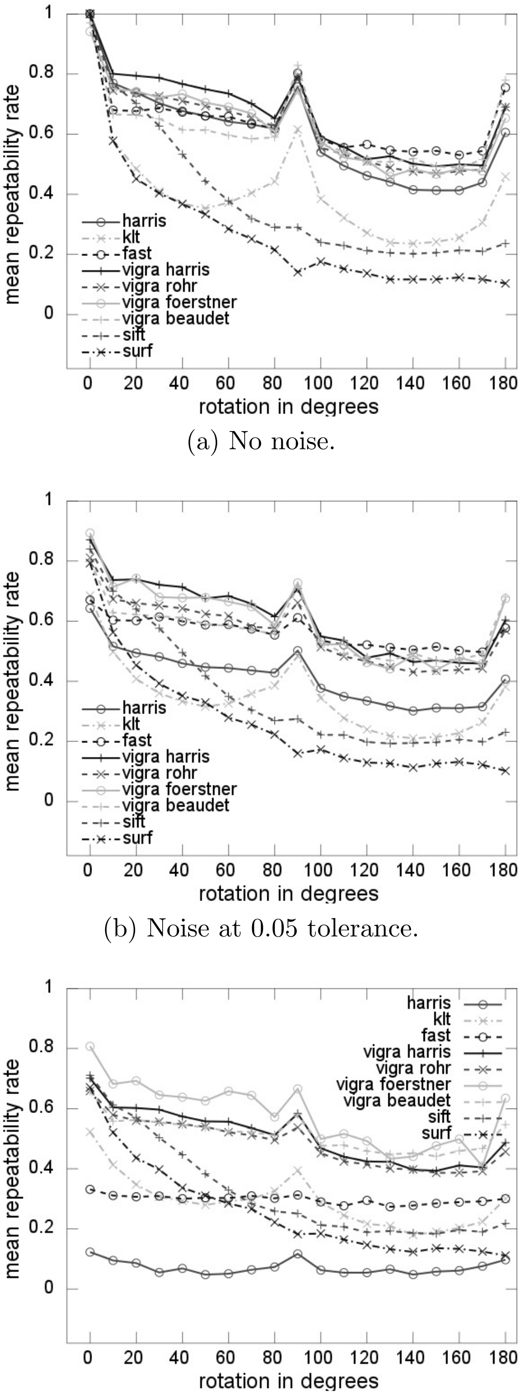


(a) No noise.

(b) Noise at 0.05 tolerance.

(c) Noise at 0.10 tolerance.

Figure 3.17: Y rotation at 1.5ϵ with all models.

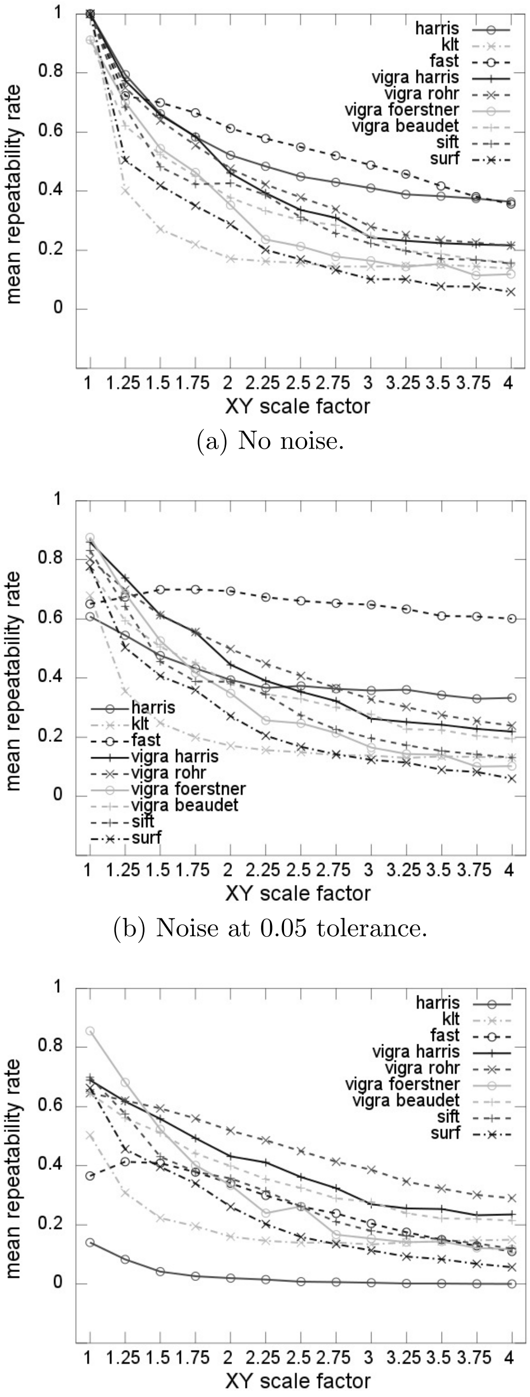


(a) No noise.

(b) Noise at 0.05 tolerance.

(c) Noise at 0.10 tolerance.

Figure 3.18: Z rotation at 1.5ϵ with all models.



(a) No noise.

(b) Noise at 0.05 tolerance.

(c) Noise at 0.10 tolerance.

Figure 3.19: Scale XY rotation at 1.5ϵ with all models.

3.5 Chapter Summary

The goal of this chapter was to investigate the establishment of a functional, reliable, flexible and accurate testing framework that can act as a baseline for testing the repeatability of interest points in a virtualised environment, as opposed to existing methodologies that utilised marked-up image sets and approximations of point positions via a homography. The goal was to be able to establish testing datasets and appropriate scene transforms that were flexible enough for testing across a range of different images and models. Section 3.2.1 proposed existing toolsets that could functionally meet these goals and could map 2D interest points to their corresponding features in 3D space. In order to properly emulate the existing repeatability methodology, affine transforms of the 3D points were utilised to replace a homography which made it possible to fully implement existing repeatability metrics with a 3D-based environment.

The feature tracking performance of nine low-level interest point detectors was systematically evaluated on Schmid's repeatability measure (Schmid et al. 2000), implemented within a 3D OpenGL virtual environment. Rotations in the X, Y, and Z axis, as well as scaling, were applied to 34 3D models, including scanned and artist-made objects, and tested with varying degrees of artificial noise. The system demonstrated was capable of finely differentiating interest point repeatability and robustness in a precise and highly reproducible way, albeit at the cost of offering only a synthetic benchmark. The results indicate a general, but not universal, superiority of the FAST detector, and should offer a useful insight into the behaviour of these interest point detectors for researchers intending to utilize them in their computer vision tasks. Results indicate that the 3D environment is able to discriminate interest point positions with a high degree of accuracy, such that very fine-grained comparisons can be made.

Other effects, such as the performance degradation of SURF, were observed. But these effects also demonstrated that the accuracy of the virtual environment's repeatability was unusually high, such that the direction of the scanning subwindow affected repeatability performance. These results show that the testing framework, STEIPR, is able to emulate not only a variety of scene transformation conditions with existing repeatability measures, but also to provide a very high level of accuracy that will ensure further tests can be considered reliable.

Chapter 4

Virtual Ground Truth, and Informedness Evaluation of 3D Interest Points for 2D Detector Repeatability

4.1 Introduction

In computer vision (CV), the establishment of ground truth so that new feature classification algorithms can be properly measured is an ongoing topic of research. With 3D scanning, printing, and realistic rendering, there are increasing opportunities for CV to be applied to virtual scenes and a multitude of new approaches are exploiting this newly-accessible niche (Guo, Bennamoun, Sohel, Lu, Wan & Kwok 2016*b*). In the field of 2D CV, there are well-accepted conventions for measuring interest-point/key-point-based feature detection, the most-well known being the work based on research by Schmid and Mikolajczyk (Schmid et al. 2000) (Mikolajczyk & Schmid 2004) which is still regularly used (Lindeberg 2015*a*), and have been used a great deal in GP research relating to interest point/key point repeatability (Gustavo & Leonardo 2006, Trujillo & Olague 2008, G. & L. 2011). It is still challenging, however, to establish a reliable means of ascertaining better ground truth of real world environments for the purposes of testing 2D-based interest point detectors (Moreels & Perona 2005) (Lindeberg 2015*a*) (Hänsch, Weber & Hellwich 2014).

This chapter[1] builds on the work done in Chapter 3, where it was demonstrated that a virtualised space, whether it be composed of images, or 3D models, served as a viable testbed for measuring interest point (IP) performance of conventional 2D detectors. Other approaches to IP generation utilise the ground truth of the model directly (Guo et al. 2016*b*), but keypoint and image descriptor classifiers, that only utilise 2D data are not designed to utilise extra dimensions. This limitation means that such classifiers can be highly optimised for 2D scenes, but not 3D, and consequently also means their performance cannot be properly measured in real-world 3D scenarios. Additionally, the lack of ground truth available for optimisation means that 3D applications for 2D-based classifiers are constrained.

With Chapter 3's demonstration of the reliability of 2D IP performance, the results from STEIPR empirically shows that virtual scenes can successfully emulate already-existing repeatability measures that are currently utilised. Given the demonstration in Chapter 3 that virtual scenes and existing methodologies can be properly integrated, this opens up the possibility for new types of testing methodologies that can be applied to the measurement of repeatability performance. With the benefit of a more robust testing environment, depth as a metric can be more accurately utilised for repeatability purposes, and it also enables the statistical classification of true negatives between 2D and 3D repeated classifications of the same IPs. Unlike existing measurements that only take into consideration type I errors, the better established ground truth that the virtual scenes afford means that type II errors can also be accurately measured. Virtual space utilisation, for testing purpose of 3D performance of type I and type II errors of 2D keypoints, has not been utilised in the field of CV research (as has been covered in the literature review of this dissertation) and now enables for more robust statistical analysis than previous methodologies. In the field of machine learning, one of the most popular forms of analysis is via ROC. Another well-known but scarcely utilised form of statistical analysis, (which has been developed more recently, and builds on ROC) is known as "informedness". Both are made viable through the application of a virtualised ground truth.

As mentioned in Chapter 2, informedness has seen a recent re-emergence in the field of machine learning, but it is non-existent in the application of CV. With the development of the virtual frameworks demonstrated in Chapter 3, ROC and

[1] These experiments were published under the title, "Virtual Ground Truth, and Pre-selection of 3D Interest Points for Improved Repeatability Evaluation of 2D Detectors", *CCVPR 2018 Conference Proceedings*, Wellington, New Zealand. https://arxiv.org/abs/1903.01828. It also received the best conference paper award at the time of presentation.

informedness can be analytically applied to the 2D and 3D performance of classifiers to provide a single-metric representation of repeatability that incorporates type I and II errors from these environments, as well as showing the best trade-off between the true positive rate and false positive rate, similar to what is provided by ROC analysis.

4.1.1 Chapter Goals

The following chapter will apply the performance evaluation of existing methodologies by Schmid et al., ROC, and informedness, that make exclusive use of the virtualised environment to more precisely analyse both 2D and 3D repeatability performance. This is possible not only due to the virtualised scene but also because of the desired deliverables of the analysis of 2D and 3D performance.

To summarise, the following research objectives will be carried out.

- Utilise the virtual scene to perform ROC performance evaluation of 2D and 3D keypoint repeatability.
- Utilise the informedness metric of the repeated keypoints to determine the best `tpr/fpr` trade-off.
- Observe the effects of integrating type II errors into the analysis of point repeatability, and identify whether the integration of type II errors differs to existing, currently-used methods.
- Establish the best training configuration that can be utilised to further analyse training performance within virtual scenes in subsequent investigations beyond this chapter.

4.1.2 Chapter Organisation

In order for this chapter to address its goals, Section 4.2 will outline how these tests will be carried out. It will be split into two types of tests, but the analysis of these two tests will be presented side by side for better comparison in Section 4.3. Both tests will be compared to better illustrate their advantages and disadvantages, as well as their relative performance in comparison to ROC, and whether it is sufficient as an analytical metric in this circumstance. Lastly, a summary of these comparisons will be made in Section 4.4.

4.2 Methodology

Schmid's metric for evaluation of a set of detectors K, classifies points between two pixel arrays $\widetilde{x}_i$, as either repeated or not, and uses a ratio of true positives and true negatives to measure performance. A threshold based on a radial distance ϵ around each point in the reference scene $\widetilde{x}_1$ determines classification. Equations 2.42, 4.2 and 4.3 describe this process, with $\widetilde{x}_1$ representing the reference scene as a basis for comparison, and $\widetilde{x}_i$ as the scene image I_i represents a member of a set of transforms j being compared. The default threshold, $\epsilon = 1.5$, represents an error rate of 1 pixel distant (as illustrated in Figure 2.4c), also known as the Moore neighborhood, and is considered by Schmid, and researchers in general that apply this metric, to be the optimal tradeoff (Schmid et al. 2000, Mikolajczyk & Schmid 2001, Mikolajczyk & Schmid 2002, Mikolajczyk & Schmid 2005). Points that do not share the same view area are removed from the validation process, as they share no valid repeatable point candidates.

For the purposes of measuring the performance of IP detectors that only use 2D images, a rendering environment that can easily switch between 2D and 3D is used. This preserves 2D consistency of detected IP classifications, while also allowing for the precision that the world space of the rendering context provides. Unlike a homography H_{1i} of the pixel positions of points within two scenes I_1 and I_i, the virtualised scene uses an inverse affine transform T_{1i}^{-1}, which enables the precise mapping of detected features to each location in world co-ordinates. Standard Schmmid-based repeatability measures utilise the pixel positions to determine whether a point is repeated or not, as well as to determine the closest point by distance (in 2D). In this case, however, when the closest point is determined before 2D-based ϵ thresholding is applied, (referred to hereafter as "preselection"), the points are represented in Euclidean space, and measured using Euclidean distances. The preselection step is described in Equations 4.5 and 4.6 now uses an 'n'-dimensional Euclidean distance function to properly represent 2D and 3D Euclidean spaces, and replaces the *dist* function to determine $R_i(\epsilon)$ shown in Equation 4.3, while not interfering with subsequent processing steps shown in Equation 4.1, which is identical to the existing approach described in section 2.9.1 and equation 4.2. Additionally, all points now include the z worldspace information (Euclidean space) as described by Equation 4.4.

$$r_{K,J(\epsilon)} = \frac{1}{N-1} \sum_{2}^{N} r_{K,I_i(\epsilon)} \tag{4.1}$$

$$r_i(\epsilon) = \frac{|R_i(\epsilon)|}{min(|\widetilde{x}_1|, |\widetilde{x}_i|)} \tag{4.2}$$

$$R_i(\epsilon) = \{(\widetilde{x}_1, \widetilde{x}_i) | dist(T_{1i}, \widetilde{x}_1, \widetilde{x}_i) < \epsilon\} \tag{4.3}$$

To enable 2D/3D preselection, D represents the vector dimensions to be utilised when measuring distance, while the function $dist$ determines the distance from the reference point in world space. Preselection happens after the removal of points that don't share the same viewport, but before the points are converted to their pixel positions and ϵ thresholding is applied. By statically pairing the closest point with its corresponding reference point in 3D space before it is measured in 2D, this preselection enables the comparison of 2D and 3D repeatability with minimal disruption, so that later analysis is simplified.

The testing configuration for 3D preselection of points follows the methodology created by Lang et al. (2014), as shown in Chapter 3. It uses a 300x300 image (I_i) which applies 47 transforms (J) of each model in the x and y axes, relative to the viewport as the model is rotated from -50° to +50° in 10° increments (11). The z is rotated from 0° to 180° in 10° increments (19), and the model is scaled in the x,y axis from 1.0, to 4.0 in 0.25 increments (17). This will be applied in two different testing scenarios. The first consists of a single model, and the other, a dataset of 12 models. Most of the models are 3D scanned, and sourced from commercial and research sites. The 12 models tested are titled "bowl", "owl", "plaque", "vase", "obelisk", "pot"[2] "marbles"[3], "apple"[4], "Stanford bunny", "happy Buddha", "dragon" and "Lucy"[5]. The "Stanford asian dragon" model is tested separately. The bowl, owl, plaque, vase, pot, apple and marbles were textured, and the rest use a generic white mesh.

$$\tilde{x_i} = (\tilde{x_{i,1}}, \tilde{x_{i,2}}, \tilde{x_{i,3}}) = T_{1i}x \tag{4.4}$$

$$T_{1i}^{-1}\tilde{x_i} = x_i = (x_{i,1}, x_{i,2}, x_{i,3}) \tag{4.5}$$

$$dist(T_{1i}, \tilde{x}_1, \tilde{x}_i) = \|x_1 - x_i\|_2 \text{ where } \|x_1 - x_i\|_2 = \sqrt{\sum_{d=1}^{D}(x_{1d} - x_{id})^2} \tag{4.6}$$

The IP detectors tested (K) were Harris (Harris & Stephens 1988), KLT (Kanade April 1991), Fast (Rosten & Drummond 2006, E. Rosten 2006), SIFT (Lowe 2004) and SURF (Herbert Bay 2006) as well as Rohr (Rohr 1992), Foerstner (Förstner

[2] http://people.csail.mit.edu/tmertens/textransfer/data/index.html

[3] http://www.sci.utah.edu/~wald/animrep/

[4] http://www.turbosquid.com/3d

[5] http://graphics.stanford.edu/data/3Dscanrep/

1986), Beaudet (Beaudet 1978) and a different implementation of Harris (Harris & Stephens 1988), which have been implemented by the Vigra library (Käthe 2000). The process for using pixel-based interest points in a pre-rendered image I_i (in conjunction with Euclidean space) to determine repeatability is similar to that which is used in Section 3.2.1, with the exception that the closest point $\widetilde{x}_i$ to the reference point $\widetilde{x}_1$ is chosen based on the Euclidean distance in Euclidean space before it is converted back to 2D co-ordinates, so that Schmid repeatability can be determined. Two different datasets were used to measure the effects of generalisation; one of these assessed repeatability at a more localised level for each transform. For the single model test, the asian dragon model was chosen due to its increased non-homogeneous surface, protrusions such as horns, and potential for misclassification of repeated points due to lack of depth, as a result of 2D preselection. This afforded an analysis based on the effects of generalisation, as well as allowing for the observation of the effects of preselection for a single model.

4.3 Results and Discussion

4.3.1 Analysis of 2D/3D datasets

When it comes to comparison of the performance of detectors, the first obvious choice is to compare the repeatability at each ϵ threshold. In most cases, a

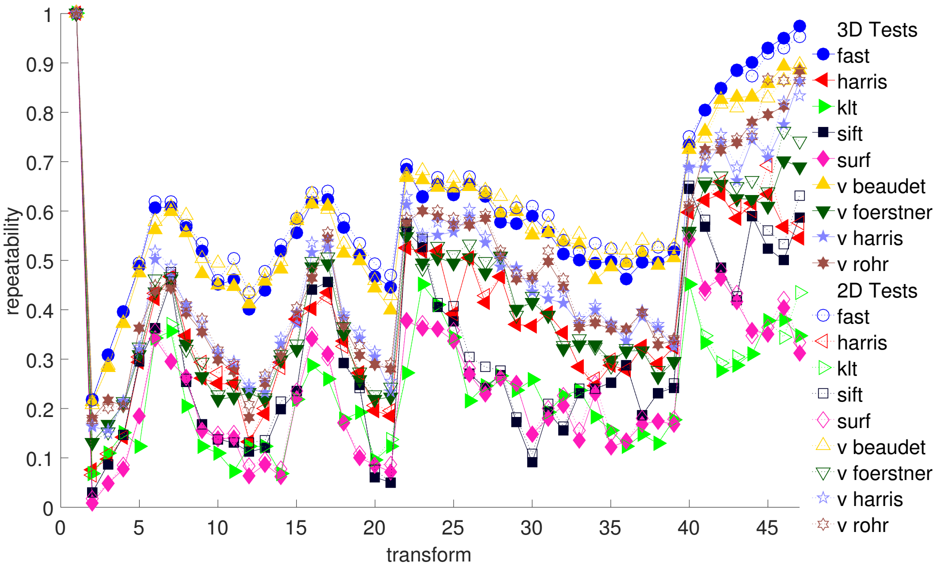


Figure 4.1: Repeatability of asian dragon model at the transform level ($\epsilon = 1.5$), with rotation in X at 2-11, Y at 12-22, Z at 23-39, and XY scale at 40-47

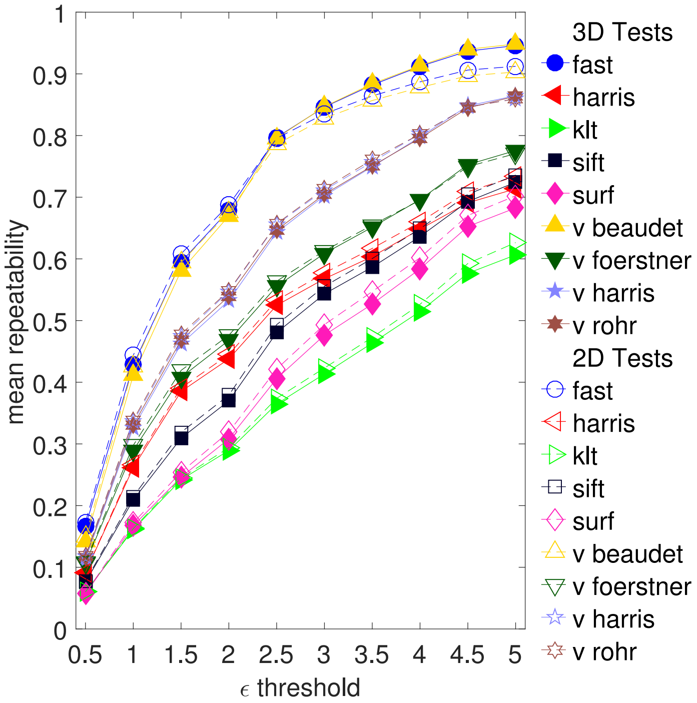


Figure 4.2: Repeatability of asian dragon model

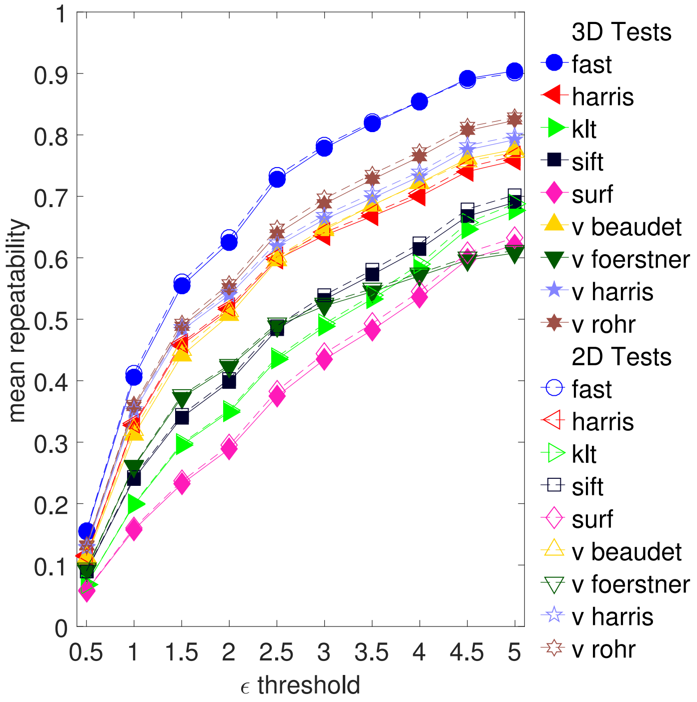


Figure 4.3: Repeatability of 12 model dataset

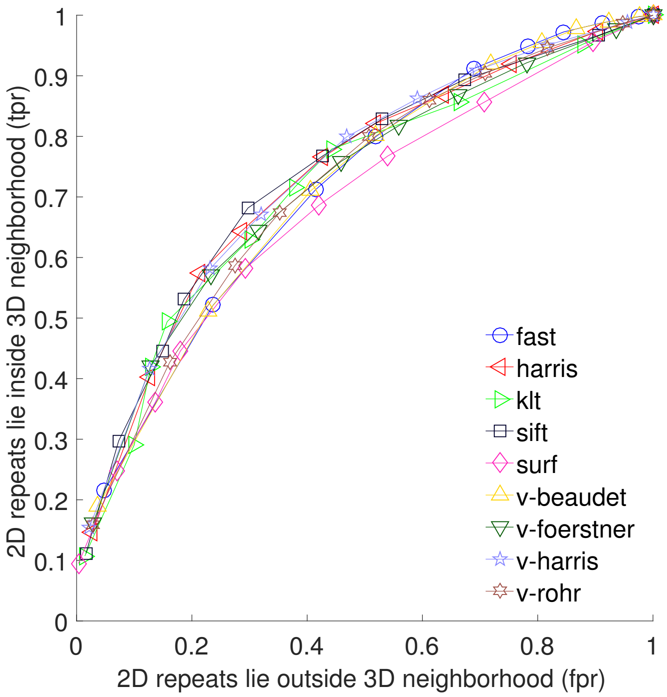


Figure 4.4: ROC of asian dragon model

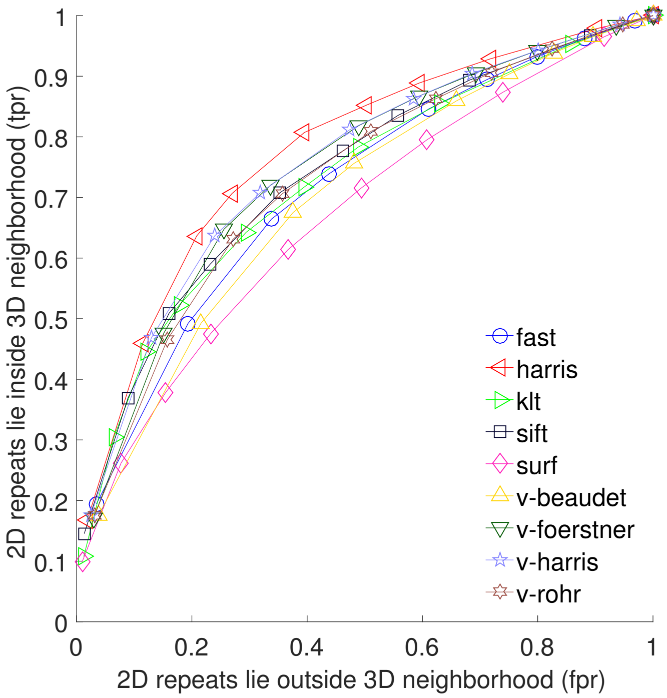


Figure 4.5: ROC of 12 model dataset

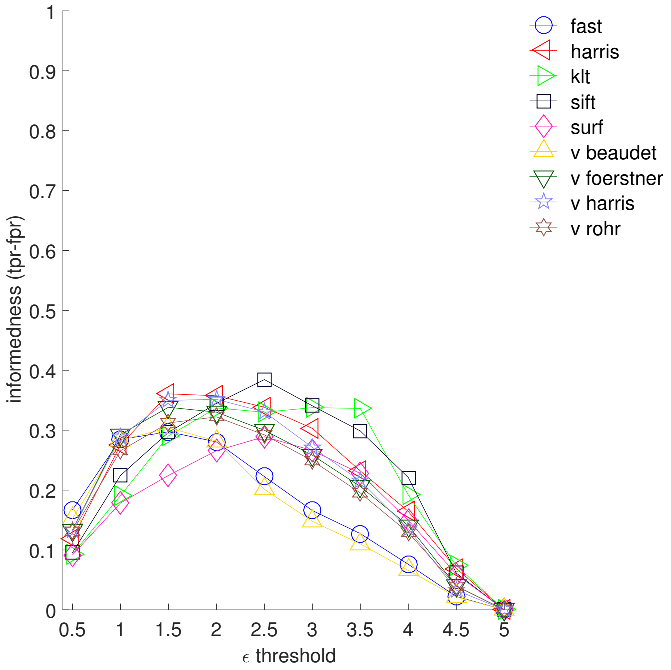


Figure 4.6: Informedness of asian dragon model

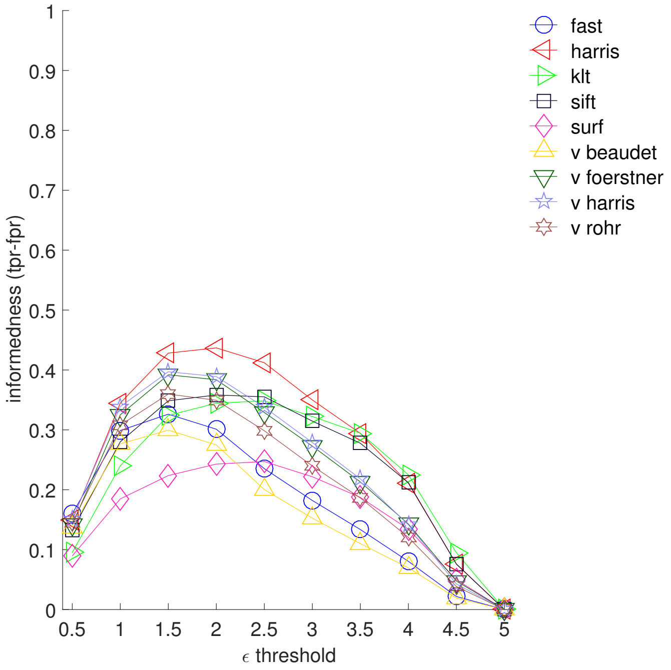


Figure 4.7: Informedness of 12 model dataset

threshold of $\epsilon = 1.5$ is the preferred threshold for discriminating between detectors. Intuitively, it would be expected that interest points that are able to utilise the depth of the scene would result in more reliable and boosted repeatability rates, given that false positives can be avoided, and better candidates chosen. The results in Figure 4.2 and 4.3 highlight that, in most cases, the 2D preselection of points provides improved repeatability performance, both across detectors, and across most ϵ thresholds. In many instances, more interest points are also detected.

At a superficial level, 2D preselection performance being better, relative to 3D preselection, could imply that 3D preselection is in fact impacting on performance, and there are indeed a few theoretical corner cases that could justify this. For example, the fact that points could become occluded, and in fact become false positives that are picked up due to their closer proximity in 2D space compared to other candidate points. Occlusion causing this type of behaviour is difficult to justify, however, as there are only a small number of the 47 transforms that could result in this type of occlusion (namely x, and y rotation of the model), and the instances of such occlusion would also require a very low number of points in order for more unusual or abnormal point candidates to be preselected. Additionally, when examining the asian dragon model at each transform (Figure 4.1), we can see that repeatability at the scene level shows the same increase for 2D preselection across all scenes. Though it is important to recognise that this is a corner case, the effect (if any) and the criteria necessary to exploit this, require exceptional circumstances.

To perform a comparison of each dataset that consisted of the single asian dragon model, and the 12model dataset, the instances of repeated point pairs between ${}_1\tilde{x}_j$ and ${}_i\tilde{x}_j$ for each test were analysed, represented as A and B respectively. To find the *tpr* (true positive rate), we intersect D_A and D_B to find true positives common to each testing configuration, and for the fpr (false positive rate), intersect and subtract the true positives. This is done at each ϵ with the equations for finding the tpr and fpr in 4.7 and 4.8. The intersection of repeated points D^3_ϵ, which represents the points that utilised 3D data, and D^2_ϵ which only used 2D data, provides a ratio of the number of true and false positives within each ϵ of a point in the reference image, which can be represented as a ROC graph.

$$tpr(\epsilon) = \frac{|D^2_\epsilon \cap D^3_\epsilon|}{|D^2_R \cap D^3_R|} \tag{4.7}$$

$$fpr(\epsilon) = \frac{|D^2_\epsilon - D^3_\epsilon|}{|D^2_R - D^3_R|} \tag{4.8}$$

This provides data sufficient for ROC analysis and calculation of an area under the curve (AUC). However, given the form of analysis that is performed in CV performance, which is to say that it is most common to compare according to the Moore neighborhood ($\epsilon = 1.5$), it is difficult to use the data in its current form for comparative analysis. Figures 4.4 and 4.5 of the same tests integrate type II errors such that it is possible to find the best performance trade-off of 2D repeated points in 3D spaces. The use of ROC also illustrates which ϵ threshold provides the best trade-off. There are some drawbacks to this type of analysis, however, as it is difficult to perform a comparative analysis. Comparing the AUC of two different classifiers needs to incorporate the results at every threshold, and as such, only provides an overall comparison. As a result, differentiating the relative performance of multiple classifiers at each threshold in a way that is similar to Schmid's methodology is not possible.

Informedness, on the other hand, provides a means of addressing these limitations by making it possible to normalise thresholds so they can be compared relative to each other as Schmid does (Schmid et al. 2000, Mikolajczyk & Schmid 2001, Mikolajczyk & Schmid 2002, Mikolajczyk & Schmid 2005). To normalise the *tpr* and *fpr* ratios for better comparison, the informedness at each ϵ threshold can be used. This provides a performance evaluation that compares both true positive and false positive detections by each classifier. Informedness is determined by finding the difference between the *tpr* and *fpr*. It has been demonstrated to be a reliable metric that can determine to a greater extent (due to incorporation of type II errors and showing the best performance tradeoffs), the similarity of data sets (compared to randomness) (Powers 2012*b*, Powers 2015*c*, Powers 2015*d*). The informedness of each detector at each ϵ threshold can be seen in Figure 4.6 and 4.7.

This new representation of the ROC data provides much better clarification of performance at specific thresholds, than AUC, while still preserving the original performance trends. In essence, informedness is ideal for this type of CV performance analysis, as it can provide an unambiguous comparison between type I performance analysis, and type I and II performance analysis in a seamless manner. Informedness also provides additional areas of performance analysis, such as establishing empirically which ϵ has the best performance trade-off relative to other classifiers, unlike Schmid's approach, which lacks this capacity.

4.3.2 Informedness Optimisation

Based on the results that are shown in figures 4.6 and 4.7, there is a clear divergence in the positions of repeated points (reflected in Schmid's repeatability approach), depending on whether preselection uses 2D or 3D point data from the same classifier dataset, even though all other testing conditions are identical. It's clear that, unlike figures 4.2 and 4.3,which use only the true positives based on Schmid's approach, there is a substantial mis-classification of repeated points depending on whether 2D or 3D data is used, that is not apparent when only true positives are taken into account. Informedness does a much better job of highlighting this disparity between 2D/3D. This should not be taken as a slight towards true positive repeatability, however, as establishing ground truth is a necessary prerequisite for such an analysis. The reality is that it is notoriously hard to reliably or accurately measure in real world environments. It does highlight that there are substantial benefits in adoption of virtualised, or more ideally, 3D scanned real-world objects, so that a more objective ground truth exists that can make these performance analyses possible.

Also of note is the fact that in the case of a singular, as well as more generalised dataset, in figures 4.6 and 4.7, the convention of $\epsilon = 1.5$, or Moore neighborhood points, is not necessarily indicative of being the most optimal convention, especially in the case of detections that are not able to preselect points with the assistance of scene depth. In fact, the informedness data suggests that $\epsilon = 2.0$ is generally more favourable across the majority of detectors when tested with the 12model dataset under the current testing conditions. This informedness of 2D detections indicates that $\epsilon = 2.0$ should be the more preferred threshold when taking into consideration the trade-offs of true positives, to false positive detections. Not only does informedness provide a more rigorous examination of 2D performance compared to 3D, but it also indicates the ϵ threshold at which 2D performance is best, which would be ideal for optimisation when it comes to taking classifiers out of the lab and into the real world. These tests demonstrate that the additional metric of informedness, in conjunction with better ground truth testing environments that can effortlessly switch between 2D and 3D, could provide new avenues of performance analysis beyond just concentrating on true positives.

4.4 Chapter Summary

This chapter explored the topic of 2D-based IP detectors and their Schmid, ROC, and informedness repeatability performance across multiple scene transformations in virtualised 3D spaces with the assistance of 2D and 3D preselection. Though there is a clear move towards utilising 3D ground truth for classifiers that use 3D ground truth natively, 2D classifiers are not able to leverage this benefit. The investigations in this chapter have sought to formulate a performance analysis that is able to integrate 3D with the assistance of a virtualised ground truth that gives a more balanced analysis of performance compared to conventional repeatability. This new performance analysis is able to provide a more balanced analysis by building on the proof of concept that virtualised 3D spaces can be used for testing 2D-based IP classifiers, and expands on this by testing the differences between finding nearest neighbor points via 2D and 3D worldspace co-ordinates, by preselecting best candidates before applying traditional repeatability metrics. Testing configurations consisted of a singular model and a 12model dataset, so as to compare the effects of generalisation, and 9 conventional 2D detectors were tested across 47 transforms in x, y and z rotation, and x,y scaling.

Though conventional 2D-based repeatability showed slightly improved performance, more in-depth analysis, made possible due to a more reliable ground truth, highlighted that 2D preselection produced a considerable rate of false positives compared to those selected using 3D. This was determined via ROC analysis, and was further refined to a singular performance metric using informedness to normalise results at each ϵ threshold. Normalisation via informedness also demonstrated that traditional conventional thresholds (eg. only including the Moore neighborhood points as repeatable; $\epsilon = 1.5$) are not necessarily optimal. The use of informedness demonstrates that other thresholds should be considered in 2D contexts for optimisation of classifiers when applied to 2D scenes, in the absence of 3D data.

From the results of these tests and approaches, it is apparent that there is a substantial difference in the repeatability, and by extension, reliability of detected points. As a metric, informedness (and its integration of type II errors) is effective at identifying the best candidate classifiers based on ϵ thresholds, as well as identifying the best performance trade-off. Informedness is well suited for applications where 2D classifiers are applied to the problem domain of 3D environments. Informedness also identified that Schmid's methodology can potentially distort how effective the performance of classifiers are when scene depth is taken into account, and give misleading analysis.

Chapter 5

Virtual Ground Truth, and Informedness Evaluation in GP: I

5.1 Introduction

In the previous two chapters, a framework for analysing 2D interest points has been developed and tested. Informedness has also been adapted to integrate type II errors discovered by using virtual spaces, with the added advantage of that it is able to empirically compare classifiers in ways that are simple and intuitive for CV, while also addressing methods for measuring 3D performance of 2D IP repeatability. Informedness has also challenged the assumption in CV that ϵ=1.5 is the best tpr/fpr trade-off in a manner similar to ROC, and that other thresholds are preferable in the context of a 2D classifier's 3D performance.

In the field of CV, Schmid's (2000) form of repeatability is well known, and used regularly for example, in GP (2.8), but given that it has been proven to provide potentially misleading analysis, as shown in the previous Chapter 4. There is a need to apply not only virtual scenes as a replacement for existing research work so that type I and type II error classifications can be properly quantified and taken into consideration during evaluation in situations where 2D classifiers are used for 3D scenes, but also to analyse these results using informedness. A field that overlaps with CV and also makes heavy use of repeatability can be found in the field of GP. As covered in Section 2.6, a great deal of GP research has depended on Schmidet al.'s repeatability approach as part of its fitness function.

Research by Olague and Trujillo (2011) makes Schmid-based repeatability a fundemental aspect of their performance evaluations, and has triggered a great deal of subsequent research that utilises fitness, with a strong emphasis on interest point repeatability, as a base for further innovation and research. As a result, informedness is a strong candidate for performing a more rigorous evaluation of repeatability in virtual scenes, as well as re-analysing existing research with a new form of statistical analysis that can seamlessly produce the same type of ϵ evaluations, and potentially replace Schmid repeatability.

In addition to re-analysing existing GP algorithms to compare it with informedness, existing GP research creates opportunities to test far more classifiers compared to the number tested in previous work, and additionally, GP serves as a testbed that can provide more flexibility, as well as enabling analysis that is statistically robust. So far, tests have only been conducted using several detectors, and the existing designs used are not able to properly remove independent variables such as the number of points, point strength, intensity thresholds and the number of nearest neighbours. Nor does it enable other factors like the use of colour channels. Employing GP testing creates opportunities to properly normalise classifier behaviour and properly compare the performance of different classifiers without the interference of independent variables present in existing keypoint and image descriptor detectors.

5.1.1 Chapter Goals

The goal of this chapter is to integrate the STEIPR system, which has been developed and tested in Chapter 3, with existing research by Olague and Trujillo (2011), in a way that enables their research to re-analyse classifier performance via not only Schmid repeatability, but also informedness. The aim is to demonstrate that evolution of interest point detectors can be performed with the use of virtual 3D scenes, based on 3D interest point repeatability data generated from 2D classifiers, to investigate how virtual scenes affect the performance of these evolved classifiers. Chapter 4 served as the groundwork for more rigorous analysis, so this chapter's focus, is the integration and testing of detectors that are evolved using GP in a virtual environment.

Given how time-consuming and processor-intensive GP is, the focus of this chapter will be on establishing the proof of concept in 2D, and then investigating which testing configurations produce favourable or viable results in the context of a greater analysis beyond this chapter. In order to achieve these aims, this chapter will build on existing configurations used in the previous chapters.

To summarise, the following research objectives will be carried out:

- Whether the STEIPR framework can be integrated with existing GP research in a seamless manner.
- Whether experiments can be closely duplicated to ensure that training of classifiers is performed seamlessly and correctly.
- Whether Informedness will allow for the measurement of statistically significant differences between different training configurations such as 2D/3D or utilisation of colour channels with different datasets.
- Whether previous informedness results that challenged the use of specific ϵ thresholds will be duplicated in GP testing.

It should also be noted that the intention is to not augment or enhance existing GP algorithms that utilise CV, but to examine how well existing GP approaches function when it is taken outside the realm of 2D datasets and introduced to 3D data that can be properly analysed. This is an important distinction, as the GP field has not currently grappled with 3D interest points or virtual 3D scenes in any meaningful way, which seems contrary to the goals of this type of research where classifiers are being optimised or simplified, 2D approximations of real-world 3D scenes in many instances. After all, why train classifiers using 2D images of real-world scenes, if the detected features in the scene cannot be measured in 3D? Chapter 4 has already demonstrated that 2D alone is insufficient and provided evidence that it results in misleading analysis. This chapter's broad intent is to investigate how the virtual environment affects GP-based optimisation approaches.

5.1.2 Chapter Organisation

The remainder of the chapter will be organised in the following manner. The first part of the chapter, Section 5.2, will focus on discussing the integration of STEIPR with GPLAB, as well as identifying and addressing possible problems that may arise due to the use of 3D scenes. The experiment setup, datasets, and methods will be discussed in Section 5.3. The results of the experiment are shown in Section 5.4 and each analysis of test comparisons will be completed collectively in this section. Section 5.5 will summarise what was found from the results.

5.2 STEIPR and GPLAB Design and Integration

Making the transition from pre-existing interest point detectors to a system that can evolve on its own has been an involved process in the writing of this dissertation. Though there is a great deal of literature on the GP systems, the undertaking of natively implementing classifiers in the form described in the literature accurately and faithfully, as well as integrating this classifier design with this dissertation's custom-built virtual environment presented a great number of technical, and functional challenges. Much of the work to this point has been setting the groundwork for applying virtual scenes in a meaningful research application, beyond just a handful of well-known detectors.

Based on the the overview of the most common forms of GP performed in CV in Section 2.6. It identified two major GP components that required implementation, and integration, the fitness function and the testing environment. STEIPR attempts to act as a replacement of the testing environment only, while providing sufficient data to satisfy the requirements of the fitness function to operate. Olague and Trujillo's implementation of the fitness function, as well as the other GP-centric operations, are handled using GPLAB. The intention of the tests is to emulate as closely as possible this earlier design, as the intention is to not modify GPLAB any more than absolutely necessary. Instead, a MATLAB wrapper function is to be utilised to co-ordinate STEIPR and GPLAB. Figure 5.1 highlights the two major testing, and fitness components, composed of GPLAB, with the wrapper MATLAB function above, and STEIPR below, along with the models/images loaded into STEIPR and represented as dataset assets, and the configurations that the wrapper function uses to interact with the STEIPR program. The system functions within a master/slave arrangement, where the wrapper function polls the STEIPR instance for data, and sends candidate classifiers for testing. STEIPR then handles the dataset loading/configuration, as well as instantiation (based on the data sent to it by GPLAB) and testing of classifiers. STEIPR is implemented in native C++, which results in the programs being separate processes. This means that multiple instances of STEIPR could be used to focus on different subsets of the dataset being tested and thus focus on minimisation of communication between GPLAB and the server instances as well as minimisation of the coordination of servers during testing. Each STEIPR instance is designed to act as a basic HTTP server, with GPLAB using basic GET requests to poll

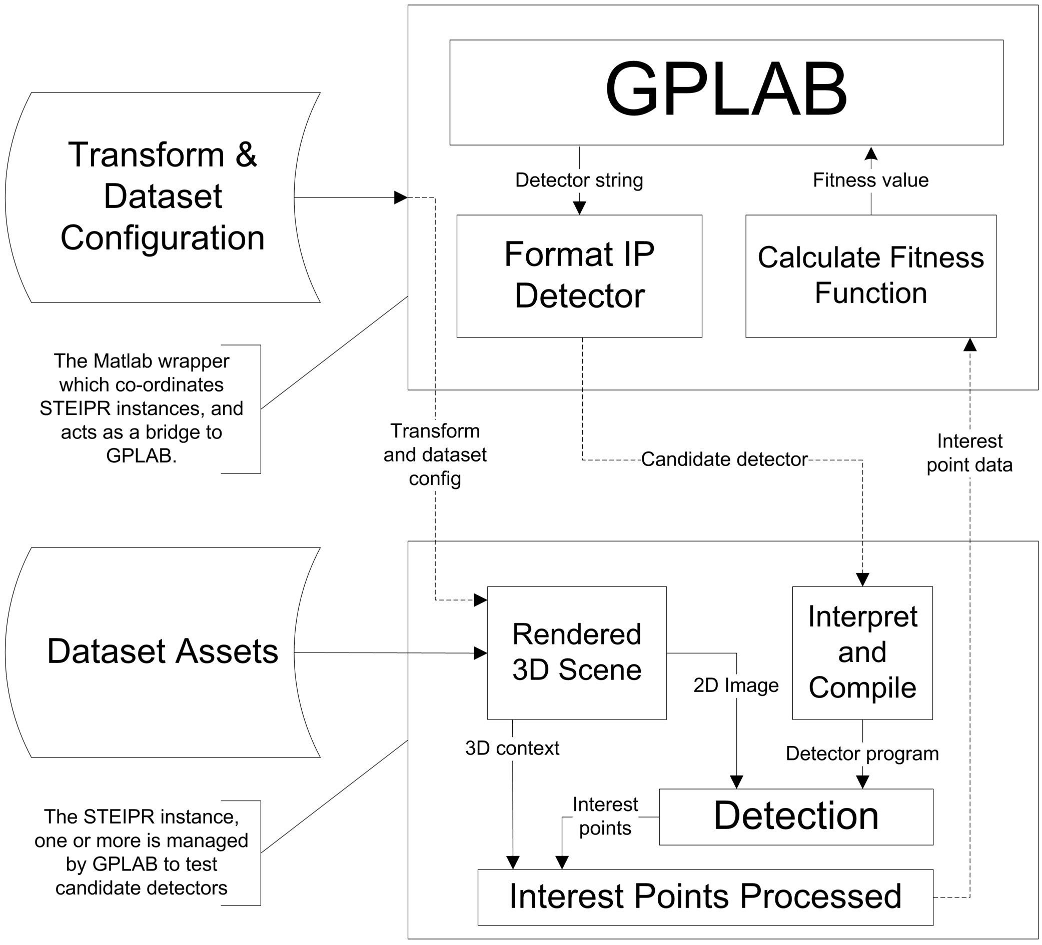

GPLAB
Transform & Dataset Configuration
Detector string
Fitness value
Format IP Detector
Calculate Fitness Function
The Matlab wrapper which co-ordinates STEIPR instances, and acts as a bridge to GPLAB.
Transform and dataset config
Candidate detector
Interest point data
Dataset Assets
Rendered 3D Scene
Interpret and Compile
2D Image
3D context
Detector program
Interest points
Detection
Interest Points Processed
The STEIPR instance, one or more is managed by GPLAB to test candidate detectors

Figure 5.1: Processing and data flow diagram of STEIPR and GPLAB.

and transmit/receive data. The messages passed between the two systems are sufficiently small that only a basic message format is required, and the complex structures required to transmit the classifier's syntax tree could be described using the JavaScript Object Notation format.

Though the design has limitations, it is still capable of handling multiple STEIPR processes to test different models concurrently. The wrapper function manages these multiple STEIPR instances. Though not strictly multithreaded, or overly efficient, it nevertheless ensures that most CPU and GPU resources are utilised as efficiency as possible. This design formed the foundations for all the evolution runs performed, and was sufficiently fast and robust that adding multiple networked machines, each with multiple STEIPR instances, was a relatively simple matter. It also ensured that larger datasets could be processed much faster than would be normally expected, and the system's simplicity meant that network communication was also robust.

5.2.1 Experimental Rationale and Explanation

It should be noted that, for the purposes of this research in this and subsequent chapters, this dissertation has focused on some of the earlier examples of GP research to test 2D and 3D IP repeatability with virtual scenes. This is deliberate and, unfortunately been misconstrued as utilising "outdated" or "superseded" methodology by some previous reviewers. As such, the use of earlier GP algorithms should be addressed and put in the correct context so that expectations are not based on certain aspects of these results un-necessarily, or unjustifiably.

What is known as the single objective GP methodology used here was originally conceived by Olague and Trujillo in 2011, with their work being refined to single and multi-objective approaches later (Olague 2016). These algorithms have been adapted to evolve image descriptors (Al-Sahaf et al. 2017), and even expanded to incorporate colour (Shao et al. 2014). As noted in Chapter 2, certain key algorithms and methodologies in the GP field have been adapted and expanded upon and have underpinned it as reliable, and well scrutinised, and as such, they are being used to test and verify the potential advantages of virtual spaces with 2D keypoints. More recent GP implementations, however, utilise the F-measure as part of the fitness function (Perez & Olague 2008, Perez & Olague 2009*b*, Fu et al. 2013, Fu et al. 2016, Al-Sahaf et al. 2017), and based on the worrying tendency of F-measure to be biased, and cause other problems (2.10.1.3), earlier GP methodologies that are well tested and known to not use F-measure have been utilised. The use of single, over multi-objective GP has also been deliberate, so as to focus primarily on repeatability without any additional independent variables potentially skewing results.

This research is not attempting to produce novel GP-centric techniques or methodologies in the strictest sense, like new fitness function approaches. Though it could be argued that this dissertation aims to produce novel GP-centric methods, this has led to confusion for reviewers of this work in the past, and it should be clarified that these tests have been employed to validate the use of informedness as a metric and virtual scenes in the broader context of the CV field, as they are applicable beyond the field of GP. The testing of 2D keypoints with virtual spaces is the primary research focus and major contributions of this dissertation. The decision to use GP in this context was based on GP's heavy usage of repeatability and its potential suitability for creating circumstances that lend greater weight to statistical significance. And though other GP algorithms exist to evolve image descriptors, dismissing evolution of 2D keypoints would be shortsighted due to the fact that the Harris detector (as highlighted in Section 2.2.6) and other 2D keypoint detectors (Wu et al. 2016) are still used in research, and, most tellingly, are being used for real-world 3D applications.

5.3 Experimental Setup

Due to the complexity of the experiments being conducted, the focus has been to use methodology that is reliable, straightforward and reusable in future experiments. These experiments have utilised pre-existing experimental methodology, including settings, software and designs. The settings used for these experiemnts are broken down as the GP-specific elements of this experimental setup (5.3.1), which covers the fitness function being utilised, the functions and terminals (5.3.2), and the parameters of the classifier's syntax tree, along with the classifier's testing configuration (5.3.3).

5.3.1 GP System Experimental Setup

As mentioned in Section 5.2.1, the type of GP duplicated as closely as possible the research completed by Olague and Trujillo (2016), and utilised only single-objective GP rather than multi-objective GP, the earlier of which is described in Section 2.8. The fitness function parameters were identical to those used by

Parameters	Description and values
Population size	50 Individuals
Generations	50
Initialization	Ramped half-and-half
Crossover and mutation probability	Crossover prob. 85%; mutation prob. 15%
Tree depth	Dynamic depth
Dynamic max depth of tree	5 levels
Real max depth of tree	7 levels
Selection	Tournament with lexicographic parsimony pressure
Survival	Keep best survival strategy
Fitness function parameters	a_x=7, c_x=5.05, a_y=6, c_y=4.3, α=20,β=20, γ=2

Table 5.1: GPLAB settings used to evolve detectors.

Olague and Trujillo (2011) however, the dataset and transforms adopted were the same as those used in Chapter 3 and Chapter 4. GPLab [1] was used to perform the necessary generation, crossover, mutation and fitness evaluation for the experiments and was run natively within MATLAB. Table 5.1 documents the GPLAB setting used.

5.3.2 Functions and Terminals

Functions and terminals have been varied slightly compared to the earlier research. Originally the histogram equalisation function was used by Olague and Trujillo (as an additional function in the function set) in earlier work on GP in the CV space. Subsequent research in the field, however, has omitted it. Due to it not having a specific functional purpose in other well-known interest point detectors, and due to its computational cost, it has been removed from the list of functions. For terminals, both greyscale (I_{grey}), and the RGB colour channels (I_r, I_g, I_b) were used, as colour has been used with some success in other GP research (Shao et al. 2014). Shao et al. utilised colour channels to optimise classifiers via GP, and colour is seeing increasing usage in the CV field (as discussed in Section 2.1.3.4 and 2.5). The functions and terminals being used in these experiments are shown

[1] http://gplab.sourceforge.net/

in Equation 5.1.

$$
\begin{aligned}
F =& \Big\{ +, |+|, -, |-|, |I_{out}|, *, \div, I_{out}^2, \sqrt{I_{out}}, log_2(I_{out}), k \cdot I_{out} \Big\}, \\
& \cup \Big\{ \frac{\delta}{\delta u} G_{\sigma_D}, G_{\sigma=1}, G_{\sigma=2}, \Big\}, \qquad (5.1) \\
T =& \Big\{ I_{grey}, I_r, I_g, I_b, L_x, L_{xx}, L_{xy}, L_{yy}, L_y \Big\}
\end{aligned}
$$

5.3.3 STEIPR System Experimental Setup

The candidate classifiers utilise a structure identical to that in Olague and Trujillo's work, but in order to optimise the processing of images, these detectors have been implemented in native C++ within STEIPR itself. This tightly couples scene processing and point processing within the STEIPR program which improves speed, but does somewhat limit flexibility and increase complexity in various ways. For the classifier to process images, the VXL [2] image library was used, as it afforded most of the functionality needed in a native C++ library. By containing point processing and scene processing, this reduced the complexity of process co-ordination between GPLAB and STEIPR. By default, the resolution of images is set at 300x300 pixels, the maximum number of points in a scene were capped at 500, the non-maximum suppression threshold is set at 255, and a nearest neighbor setting is $n = 2$. Before max points is applied, the interest points are sorted from strongest to weakest. The 300x300 image size deviated from the Olague and Trujillo's original tests slightly, but the processing of images rose considerably with larger images, and this was deemed to be a sufficient compromise to keep the training time within reasonable limits.

5.3.4 Dataset and Transform Experimental Setup

To demonstrate the proof of concept, an identical testing methodology to that in Chapter 4 was used. This resulted in 47 transforms being processed for each image. The models used were rotated in the X, Y axis from -50° to 50°, in the Z axis from 0° to 180° clockwise, and scaled in the X,Y axis from 1.0 to 1.25, 1.5, 1.75, 2.0, 2.5, 3.0, 3.5, 4.0. Appendix A.5 shows the raw data used to generate

[2] http://vxl.sourceforge.net/

Parameters	Description and values
Max suppression nearest neighbor	2
Viewport	300x300
X rotation	-50° to +50° in 10° increments
Y rotation	-50° to +50° in 10° increments
Z rotation	0° to +180° in 10° increments, clockwise
X,Y scale	1.25, 1.5, 1.75, 2.0, 2.5, 3.0, 3.5, 4.0
Texture filtering	Anisotropic and linear
Max points returned	500

Table 5.2: STEIPR settings.

the terminals for each of these transforms. The same image from Chapter 3 was used. Based on previous tests, it was decided that the transforms were sufficiently complex to test repeatability without so much occlusion or distortion to the model that the features would not be repeatable. Though previous research by Olague and Trujillo used far fewer transforms, we felt that more rigorous testing and analysis would be feasible, given the system's processing capacity. The viewport of the scene was set at 300x300 pixels, and used anisotropic and mipmap nearest texture filters. This is shown in Table 5.2.

The datasets used for training and testing were reused from previous experiments. The sower image from Chpter 3, the asian dragon model, and 12model dataset from Chapter 4 will be utilised to investigate the advantages/disadvantages and certain characteristics of training in virtual environments. Each dataset used the same scene transforms and settings as those shown in Table 5.2.

5.3.5 Training and Testing Configurations

The initial configuration being tested serves as a proof of concept to demonstrate that the system can optimise classifiers based on the parameters chosen. The first test utilised the sower image in both 2D and 3D to demonstrate the proof of concept. Given that the image should make little difference whether the classifier is trained in 3D or 2D, it should serve as a benchmark that classifiers optimised in a 3D environment perform on par with the conventional 2D approach. To build on the use of the 3D-trained classifiers, greyscale and RGB were trained and compared. This has the added benefit of demonstrating that informedness can be utilised to examine the performance trade-offs in areas beyond just depth. In both cases, the 12model dataset used in the previous chapter was used to test the performance of these evolved classifiers. The larger testing dataset was intended

Training dataset	Testing dataset	Color	Training runs	Light position	Image size	2D/3D	Max points
sower	12models	GS	30	center	300px	2D	500
sower*	12models	GS	30	center	300px	3D	500
sower	12models	RGB	30	center	300px	3D	500
12models	sower	RGB	30	center	300px	3D	500
12models†	sower/dragon	GS	50	center	300px	3D	500
12models	dragon	GS	50	center	300px	2D	500

Table 5.3: Testing and training configurations. Note "*" denotes that both the training and testing data were reused without modification or retraining for each analysis in Table 5.4, and "†" denotes that only the training data was reused, and that the analysis used a different testing dataset in each case.

to provide sufficient generalisation. It was unknown whether utilising a single scene for training would result in a degradation in performance compared to the results that would be achieved using conventional detectors, however. To test this, the reverse was carried out to determine whether utilisation of larger datasets for training would result in better performance outcomes when compared to the results achieved when using a single scene.

In total, based on the planned testing and training requirements, this results in 8 training and 8 testing configurations, as summarised in Table 5.3. The number of training runs per configuration varies between 30 and 50. The factors affecting the number of runs are priority and time. Some of these training datasets have been reused in across different analyses, due to their similarity in settings. For example GS training and 3D training have been reused, but also due to our desire to make better utilisation of more time consuming training runs compared to others, in particular, training with the 12model dataset took far longer than training with a single model/image. For the analysis of the testing results, an approach similar to that in Chapter 4 was performed to directly compare the Schmid repeatability and informedness performance of each training/testing configuration's best evolved classifier for each training run. It should be noted that 2D and 3D training should not be confused with the 2D and 3D (or RGB and greyscale) data that is used by informedness during testing. In order to simplify the analysis and organisation of results, each training/testing configuration being compared was labeled alphabetically. The analysis of training runs is summarised in Table 5.4.

Analysis	Training dataset	Testing dataset	Color	Training runs	Light position	Image size	2D/3D	Max points
A	sower	12models	RGB	30	center	300px	3D	500
	sower*	12models	GS	30	center	300px	3D	500
B	sower*	12models	GS	30	center	300px	3D	500
	sower	12models	GS	30	center	300px	2D	500
C	12models	sower	RGB	30	center	300px	3D	500
	12models†	sower	GS	50	center	300px	3D	500
D	12models†	dragon	GS	50	center	300px	3D	500
	12models	dragon	GS	50	center	300px	2D	500

Table 5.4: Sorting of analysis of training runs. Note "*" denotes that both the training and testing data were reused without modification or retraining for each analysis in Table 5.4, and "†" denotes that only the training data was reused, and that the analysis used a different testing dataset in each case.

5.3.6 Experimental Hardware, Training Load and Timeframe

To conduct the majority of the training, and subsequent testing, we used three machines, one with lower-end hardware and two higher-end hardware. The first lower end machine consisted of an Intel i7-4770k @ 3.5GHz CPU, 32GB of RAM and a NVIDIA GeForce GTX 780 with 3GB RAM, running on a Windows 7 OS. The second machine had an Intel i7-6700k @4.0 GHz CPU, 32 GB of RAM, and a GeForce GTX 1080 GPU with 4GB RAM, also on a Windows 7 OS. The third machine had an Intel i7-8700k @3.7GHz CPU, 32GB of RAM and a GeForce GTX 1080 GPU with 8GB of RAM, running on a Windows 10 OS. Collectively, these machines could run 12 STEIPR processes at full load (2 processes on the low-end machine and 4 and 6 each on the other two). During testing, the GPU load on each machine varied between 60 and 80% usage of its cores at full CPU load. The STEIPR process' RAM usage, depending on the dataset used, varied from several hundred MB for the sower, to 2-3 GB per process for the 12model dataset.

The time required to process a single scene according to the STEIPR settings shown in Table 5.2 and the GPLAB settings shown in Table 5.1 is 8 hours on average and is constrained to a single CPU core. For tests such as the sower, or asian dragon model, the processing times were largely identical regardless of the model complexity. Larger numbers of models increased the time linearly, such that under identical conditions with the 12model dataset, the time required to finish a single training run was 96 hours on average. With 12 scenes, this also increased the variance in training time considerably by ±30%, and was generally caused by the classifier syntax tree's occasional overuse of computationally expensive functions. Though in rare cases, training time could take as long as 50% longer

Dataset	CPU hours per training run (avg.)	Total training runs	Total CPU hours	Time required with 12 concurrent processes	
				Hours	Days
sower	8	30	240	20	0.83
12models	96	30	2880	240	10
12models	96	50	4800	400	16.66

Table 5.5: Approximate training time required under ideal conditions.

than the average time to finish a single training run where the use of the 12model dataset was concerned. Under theoretically ideal conditions, with the training hardware mentioned previously, 30 training runs could be completed within a day with a single model. However, in practice, the manual co-ordination and handling of this process meant that the process could be considerably slower due to technical complications. The ability to run 12 STEIPR processes concurrently afforded a considerable increase in the theoretical training times, but efficient use of all system resources for training was limited by the fact that manual monitoring was still required for failures, completion, and the running of new tests. These best case scenario time frames are highlighted in Table 5.5.

During training, checkpointing at each generation was used to offset failed runs. The failure rate was approximately 3%, which facilitated manual intervention, resolution and resumption of training at the most recently-checkpointed generation where failure occurred. The nature of failures ranged from network connectivity issues between the STEIPR and GPLAB processes, to GPU crashes and CPU overloading resulting in freezing, or disk paging. In order to remove disk usage bottlenecks, no logging of raw data was performed during training. Instead, raw data logging was performed as an additional step after training was finished, and only for the last generation trained. This increased the processing of results by approximately 1/10th of the time spent training, as the last generation of the training run had to be re-tested and logged to disk. Performing raw data logging as an additional step was deliberate, as it avoided unnecessary logging, which would also have impacted on other training being carried out. All logging was done on a separate, non-training machine to avoid interference.

5.4 Results and Discussion

The results of these tests are intended to investigate the viability of optimising keypoint classifiers via GP algorithms, and test trained classifier performance via Schmid repeatability and informendness in a comparative manner. It has already been demonstrated that direct comparisons can be made between both Schmid repeatability and informedness, and so the purpose of this discussion is to further explore the testing configurations proposed and determine whether these configurations impact on performance in a statistically significant way. To give a broader overview of the performance of classifier evolved based on Olague and Trujillo's algorithms, tests of conventional detectors, as seen in Chapter 4 are included in this discussion for informational purposes only. Unfortunately, due to the nature of the conventional detectors used in previous tests, there was no simple or effective means to cap the number of points so as to ensure a reliable comparison of performance metrics, as most of the detectors did not return point strengths that could be ordered. As such, lack of point capping could affect repeatability in unpredictable or unintended ways but these results are included here nevertheless where applicable as they only utilise RGB. In Chapter 6, this will be further explored.

The discussion of this chapter is focused on isolating and testing specific characteristics of the virtual space to determine with confidence whether a specific configuration aids or hinders performance. This means that there will be less emphasis on GP-centric analysis like fitness, or the classifiers themselves at an individual level, as these are beyond the scope of this dissertation. The emphasis here is on analysing the utilisation of virtual scenes, and the advantages, or lack thereof, of informedness compared to Schmid.

5.4.1 Analysis A

Figures 5.2, 5.3 show the box plot performance of all runs for analysis A, and Table 5.6 shows the F- and p-values of these results for Schmid repeatability and informedness. The configurations aim to compare the use of RGB terminals or a simple greyscale (GS) terminal for optimisation on the sower image, with subsequent testing with the 12model dataset. It should be noted that the 12model dataset only consists of half the models using any colour, as shown in Appendix A.2, and given that training was performed on the sower image which uses a

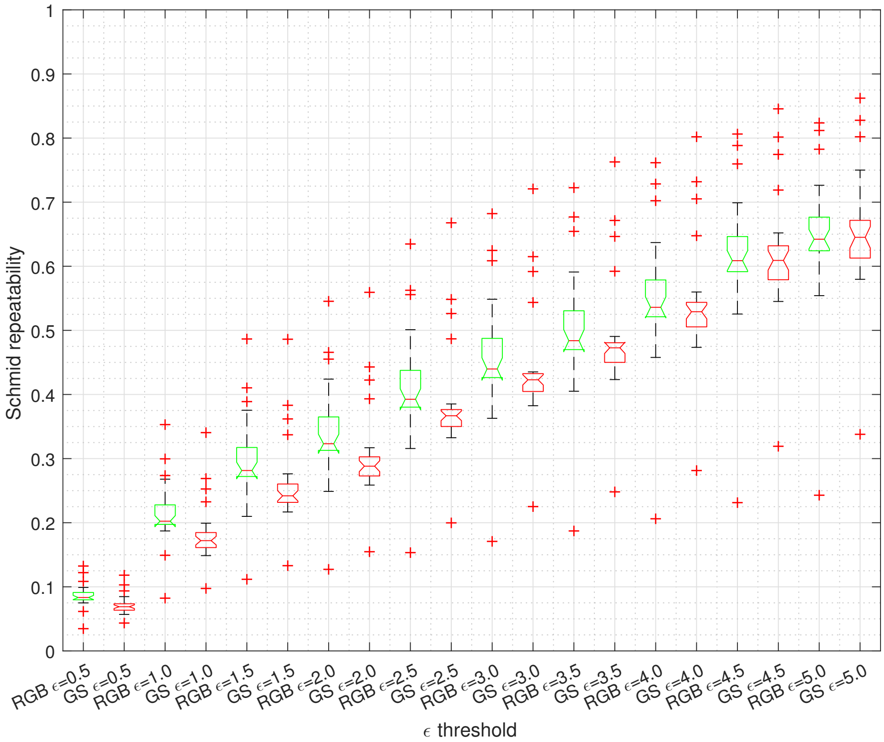


Figure 5.2: Analysis A: RGB(green box)/Greyscale(red box) training with sower with 3D, and Schmid repeatability tested using 12models dataset. 30 training runs each.

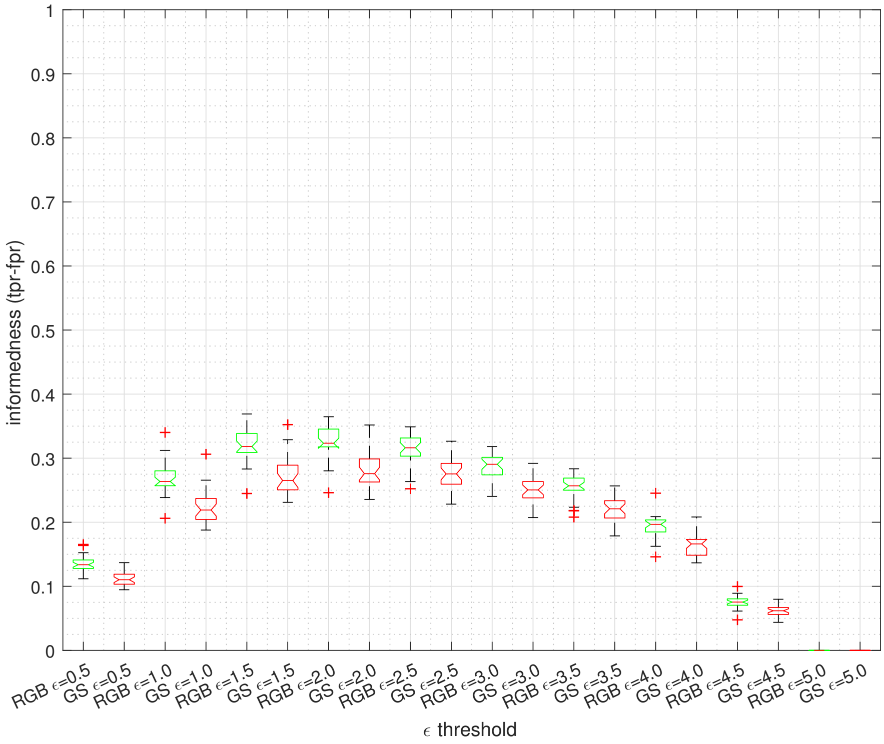


Figure 5.3: Analysis A: RGB(green box)/Greyscale(red box) training with sower with 3D, and informedness tested using 12models dataset. 30 training runs each.

large variation of colour, subsequent testing would not necessarily predispose performance towards classifiers that specialise with colour terminals, and would in fact make it more difficult to specialise in colour in this case. With a significance level of 0.05, Schmid shows a statistically significant divergence in ranges from $\epsilon = 0.5$ to $\epsilon = 2.0$. Informedness, however, shows an even stronger confidence at nearly every ϵ. The informedness trade-off also peaks at $\epsilon = 2.0$. This demonstrates that even beyond 2D/3D applications, informedness was capable of utilising virtual scenes to analyse performance trade-offs, as well as confirming the analysis of performance differences under different conditions, in this case, the advantages of utilising colour over greyscale. The larger testing dataset (12models) also aided in providing generalisation and its composition of both colour-based and greyscale models helped prevent any particular bias towards colour optimisation. As shown in Chapter 4, the informedness measure, made possible through the use of the virtual scene's better ground truth, provides an empirical analysis of the best performance tradeoff, which is not evident in Schmid repeatability. This test and setting configuration also demonstrates that $\epsilon = 2.0$ is the preferable threshold, rather than the more standard $\epsilon = 1.5$. Table 5.6 also shows that at $\epsilon = 2.0$, the (F-value=4.4084) exceeds the F-critical threshold 4.0069 , and the p-value=8.10-E009, well below the α level of 0.05.

Analysis A: RGB/GS train: sower, test:12models (30 runs each)				
	Schmid repeatability		Informedness	
ϵ	F>4.0069	p-value (α=0.05)	F>4.0069	p-value (α=0.05)
0.5	14.163	0.00039298	59.779	1.73E-010
1.0	8.1138	0.0060689	46.389	6.08E-009
1.5	6.017	0.017197	46.689	5.59E-009
2.0	4.3084	0.042369	45.385	8.10E-009
2.5	2.1995	0.14347	40.677	3.22E-008
3.0	1.2228	0.27337	45.548	7.73E-009
3.5	0.68308	0.41192	45.146	8.67E-009
4.0	0.20193	0.65484	40.856	3.05E-008
4.5	4.16E-005	0.99488	30.206	9.08E-007
5.0	0.032449	0.85767	NaN	NaN

Table 5.6: F- and p-value at each ϵ for analysis A.

5.4.2 Analysis B

Figures 5.4, 5.5 and Table 5.7 show the box plot performance of all runs for analysis B, as well as the F- and p-values of these results for Schmid repeatability

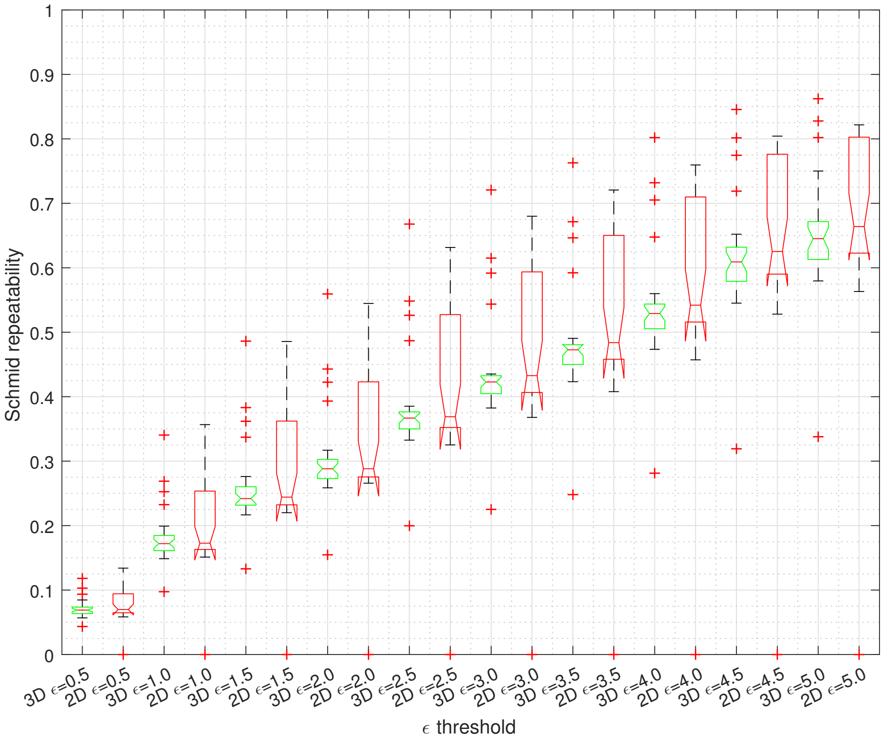


Figure 5.4: Analysis B: 2D(red box)/3D(green box) training with sower, and Schmid repeatability tested using 12models dataset. 30 training runs each.

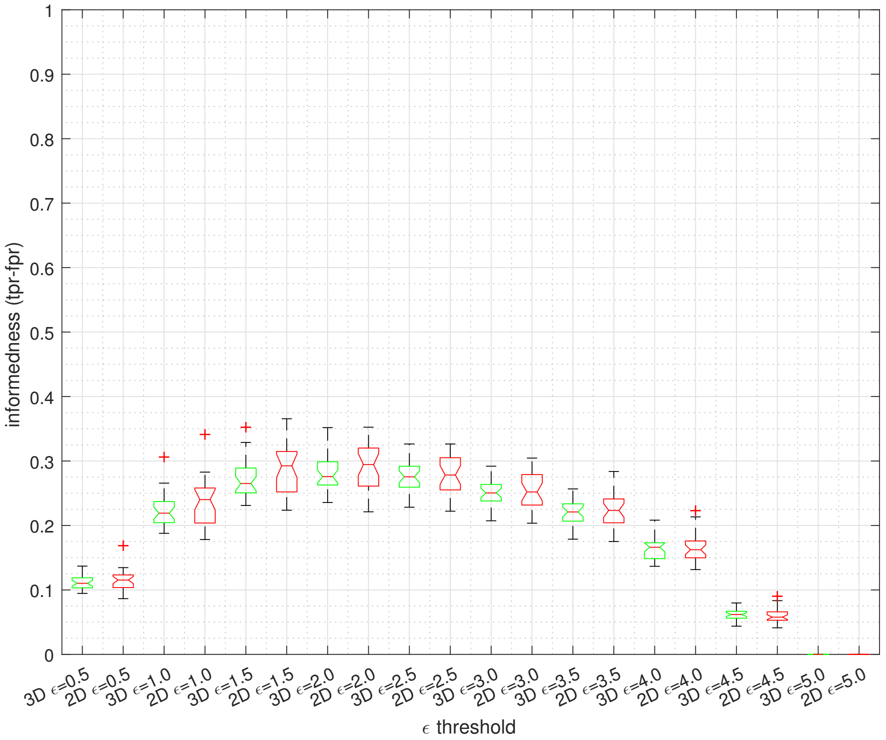


Figure 5.5: Analysis B: 2D(red box)/3D(green box) training with sower, and informedness tested using 12models dataset. 30 training runs each.

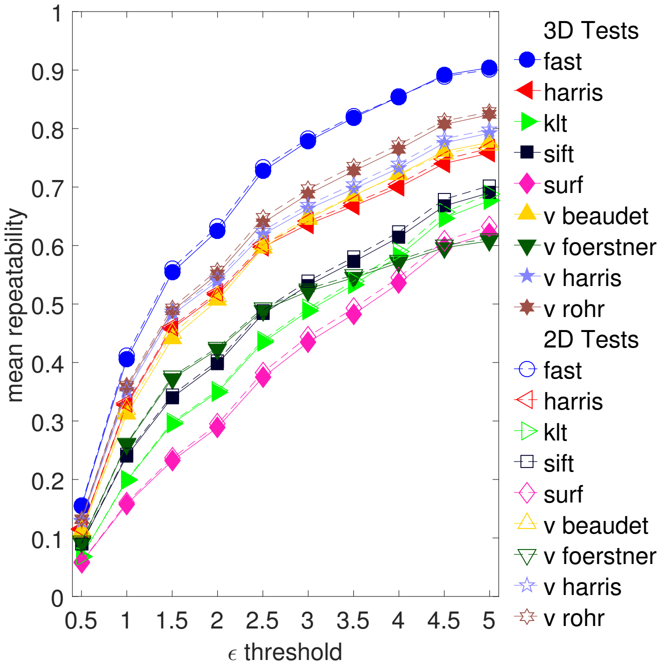


Figure 5.6: 2D (open-dashed) overlaid versus 3D (solid lines) Schmid repeatability of 12model dataset, as seen in Chapter 4.

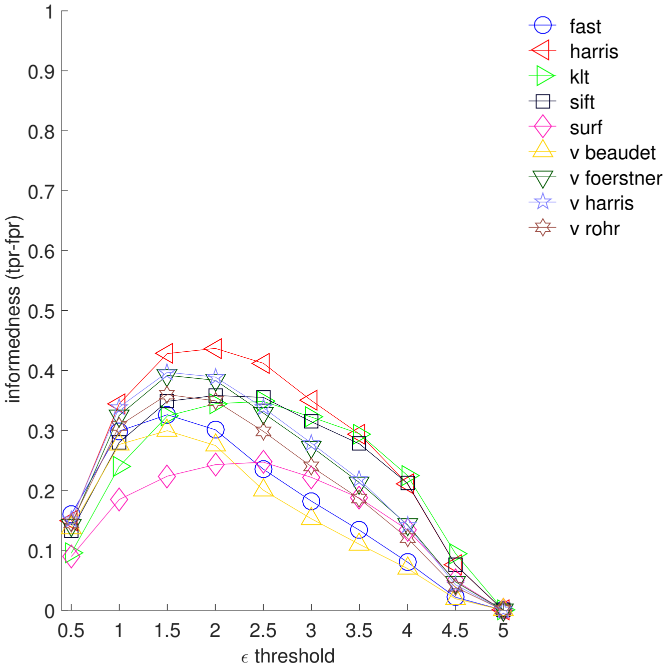


Figure 5.7: Informedness of 12model dataset, as seen in Chapter 4

and informedness. The objective of this analysis is to investigate the differences of training with the utilisation of 2D and 3D IP repeatability. However, whereas Chapter 4 utilised 3D models, this test uses a flat pane in 3D space, as used in Chapter 3. This analysis reuses the training and testing data from the previous sower image greyscale training and testing in analysis A, in order to determine whether any optimisation that is statistically different is afforded in comparison to 2D. In this analysis, the expectation is that there should be no repeatability difference between the best 2D and 3D candidates of each training run. The test is intended to demonstrate that a 2D surface in a virtual space will behave identically, even when 3D data can utilise 3D data of the scene to measure repeatability, and should afford no difference in optimisation. It should also demonstrate that the use of 3D does not degrade performance either, and that the optimisation of classifiers functions equally well in both circumstances. In Figure 5.4 the means and notches of both groups at each ϵ threshold do not show any apparent divergence visually. However, the third quartile of the 2D data is skewed higher than 3D data which shows a much tighter interquartile convergence on the mean, albeit with more outliers past the maximum. In both Schmid and informedness, there is no apparent divergence beyond the notches, and visually the box plots do not indicate any statistically significant difference between 2D and 3D medians at each threshold. This confirms what was expected, and demonstrates that though the Euclidean space is being fully utilised to measure distance (in comparison to 2D space, which only used 2D data of the scene), the performance in both cases is the same, with 95% confidence. Analysis of Schmid repeatability and informedness across all ϵ ranges do not show any significant divergences in the

Analysis B: 2D/3D train sower, test: 12models (30 runs each)				
	Schmid repeatability		Informedness	
ϵ	F>4.0069	p-value (α=0.05)	F>4.0099	p-value (α=0.05)
0.5	1.0724	0.3047	0.82635	0.36716
1.0	1.6596	0.20277	1.7408	0.19232
1.5	1.76	0.18982	1.4423	0.23474
2.0	1.7601	0.18982	1.2642	0.26557
2.5	1.6897	0.19878	0.77405	0.38266
3.0	1.5563	0.21722	0.67068	0.41623
3.5	1.3929	0.24274	0.18041	0.67262
4.0	1.1182	0.2947	0.06644	0.79752
4.5	0.72409	0.3983	0.19684	0.65896
5.0	0.5443	0.46363	NaN	NaN

Table 5.7: F- and p-value at each ϵ for analysis B.

minimum and maximum of each population of best classifier candidates. The informedness box plot's third quartile in Figure 5.5 does show a slight preference towards 2D, but that was not significant. Figure 5.5 mean performance also shows the best trade-off in both 2D and 3D peaks at $\epsilon = 2.0$, though the 2D and 3D maximum peaks at $\epsilon = 1.5$.

The p-values and F-values shown in Figure 5.7 do not indicate any divergence in populations either. The slight variance in the third quartile of Figure 5.4 could have been due to the number of training runs performed; however, there is already a strong convergence of the confidence intervals in the box plots as well as very similar maximum and minimum ranges in all cases. Though we cannot make any strict assumptions about performance, the figures 5.6 and 5.7 show conventional detectors tested under the same conditions (albeit with no max point limit). For Schimid, the Fast detector has a stronger repeatability, and for informedness, Harris shows slightly better performance. The other detectors perform as well, or worse than the best-evolved classifiers. Also of note is that the conventional informedness trade-off peaks at $\epsilon = 2.0$ for Harris.

5.4.3 Analysis C

Figures 5.8 and 5.9 and Table 5.8 show the box plot performance of all runs for analysis C, along with the F- and p-values of these results for Schmid repeatability and informedness. Analysis C is similar in configuration to Analysis A (5.4.1) but aims to reverse testing and training datasets to the 12model dataset from training and the sower image for testing. In Figure 5.8, Schmid repeatability shows no discernable difference in the mean at each ϵ threshold, as well as substantial numbers of outliers. Figure 5.9 shows a great deal of variation in informedness, as well as poor overall performance with no definitive best trade-off. This is to be expected for a 2D image no matter how it is rotated, due to the fact that it is not able to create situations where false positive detections, as shown in Figure 5.11, though in the case of scaling, the resolution can create more unstable points, and produce false positives in that manner. Examination of the raw informedness data also reveals that many values were in the "perverse" (2.10.1.1) range according to the ROC data, mainly due to the fact that there were barely any false positives (cumulatively across all model transforms) compared to tens of thousands of true positives. This resulted in ROC data that was not sufficient for analysis, and though this may seem unusual, it should be expected that testing the 3D/2D informedness performance of classifiers with a 2D model will struggle to misclassify

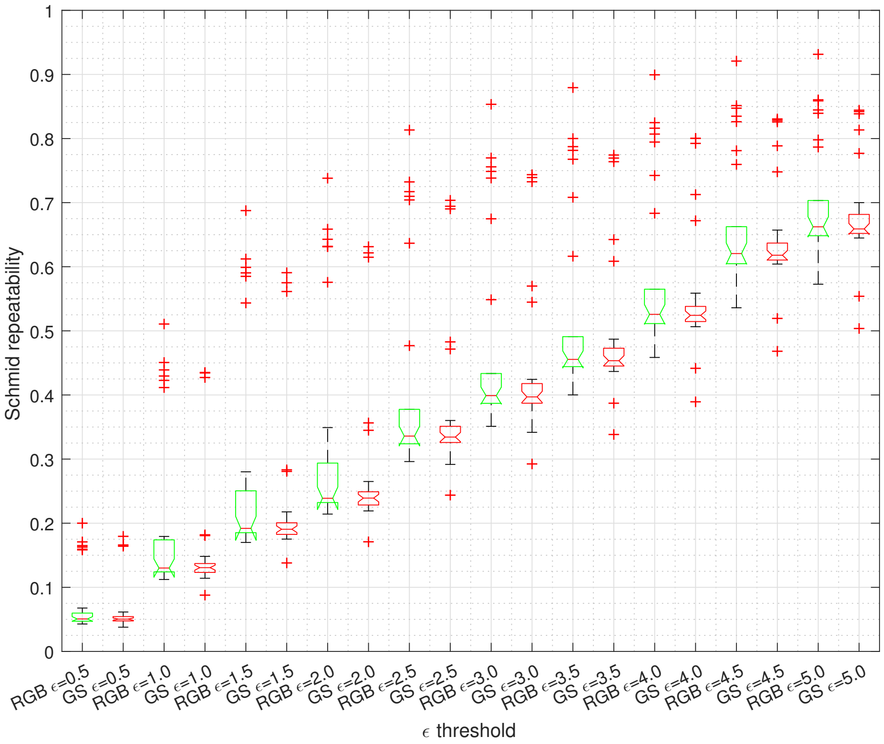


Figure 5.8: Analysis C: RGB(green box)/Greyscale(red box) training with 12models with 3D, and Schmid repeatability tested using sower. 30 trained using RGB, and 50 trained using greyscale.

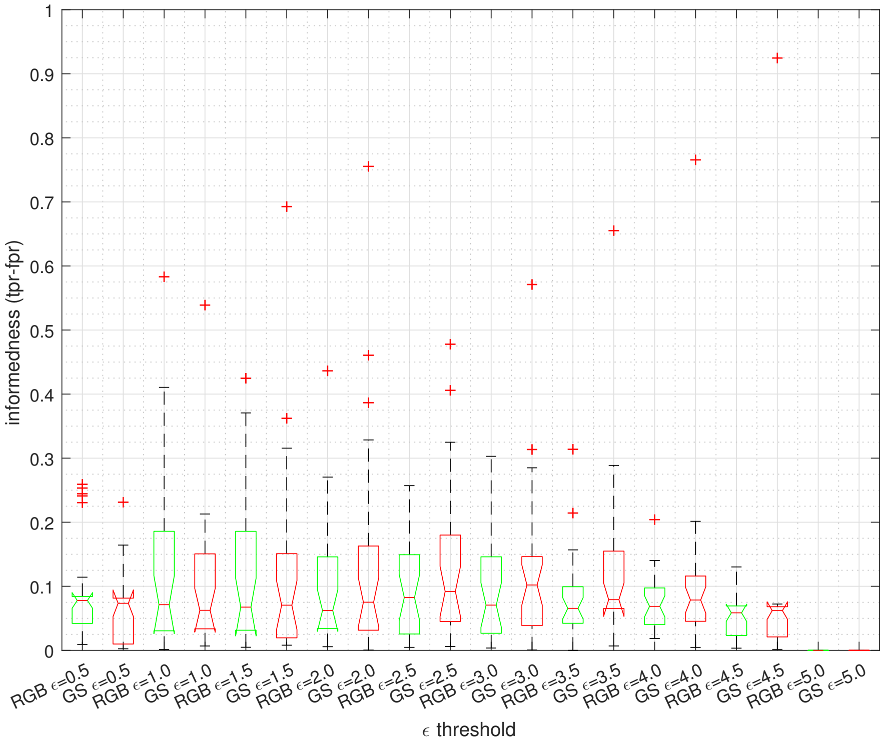


Figure 5.9: Analysis C: RGB(green box)/Greyscale(red box) training with 12models with 3D, and informedness tested using sower. 30 trained using RGB, and 50 trained using greyscale.

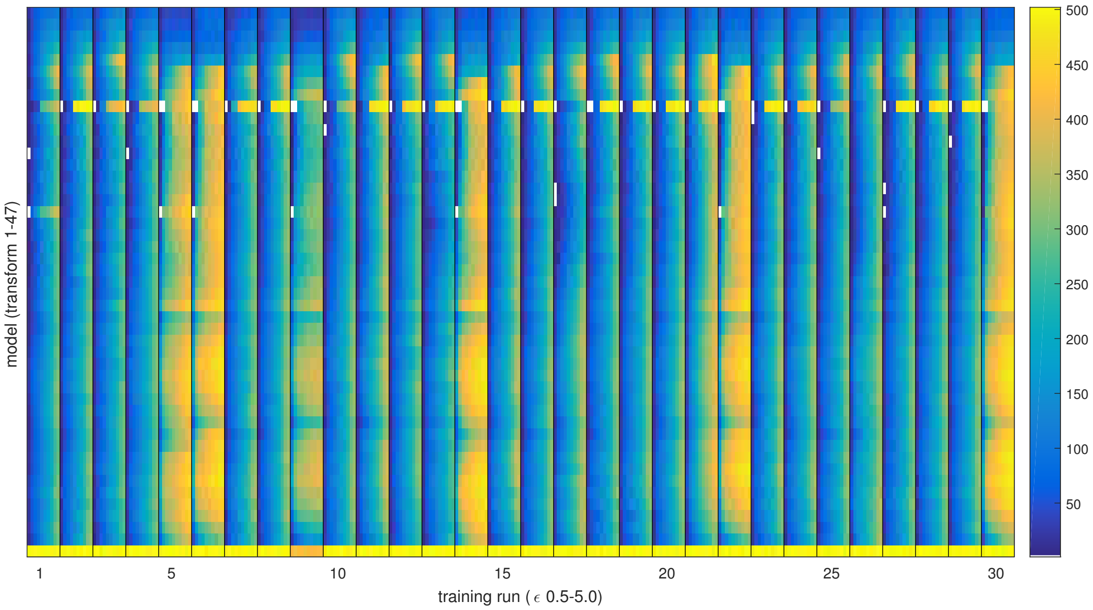


Figure 5.10: Analysis C: Heatmap of true positives (TP) repeated points for sower testing, 30 training runs 12models 3D RGB.

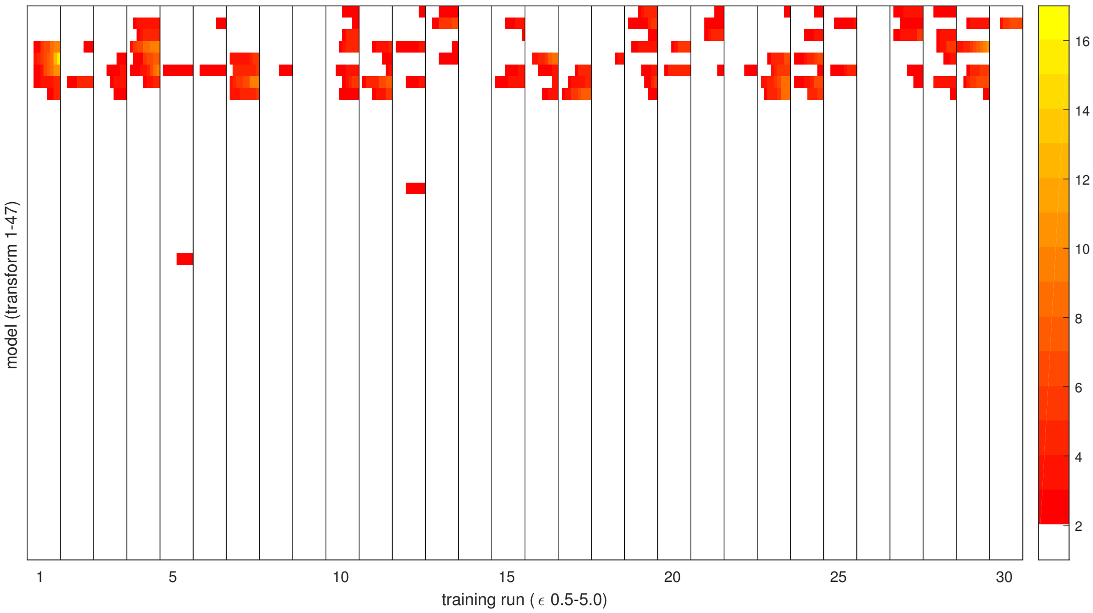


Figure 5.11: Analysis C: Heatmap of false positives (FP) repeated points for sower testing, 30 training runs 12models 3D RGB.

features in 3D. Of course, because Schmid does not discriminate classification collectively based on 2D/3D performance, it remains unaffected. Figure 5.11 illustrates the 30 training runs tested with the sower image, and presents the numbers of false positive repeated points when each classifier is tested with the sower image and compared to the TP in Figure 5.10.

Analysis C: RGB/GS train: 12models, test: sower (30/50 runs)				
	Schmid repeatability		Informedness	
ϵ	F>3.9635	p-value (α=0.05)	F>3.9685	p-value (α=0.05)
0.5	1.3964	0.24092	2.0123	0.16017
1.0	1.5973	0.21006	0.30748	0.58088
1.5	1.7452	0.19034	0.36645	0.54677
2.0	1.7145	0.19425	1.7883	0.18518
2.5	1.7335	0.19182	3.3647	0.070573
3.0	1.6516	0.20254	3.4149	0.068551
3.5	1.6659	0.20062	4.6895	0.033527
4.0	1.6815	0.19855	2.3189	0.13202
4.5	1.6777	0.19904	1.5343	0.21933
5.0	1.6429	0.20373	NaN	NaN

Table 5.8: F- and p-value at each ϵ for analysis C.

5.4.4 Analysis D

Figures 5.12, 5.13 and Table 5.9 show the box plot performance of all runs for analysis D, along with the F- and p-values of these results for Schmid repeatability and informedness. The purpose of this analysis is similar to analysis B, except with the difference that this analysis aims to properly measure potential improvements in 2D/3D IP performance when training with a 3D model, as opposed to a flat pane where 3D demonstrated no statistically measurable difference. The expectation was that the utilisation of a larger dataset, 12models, and a higher number of training runs (N=50) would aid in producing results that could be definitively examined. Based on the Schmid and informedness comparisons of the the 2D and 3D training, Figure 5.12 shows no statistically measureable difference in mean performance. Informedness of the 2D/3D evolved classifiers also confirms this in Figure 5.13. Table 5.9 confirms that both F-, and p-values are within 95% confidence that the 2D and 3D performance is the same. Comparatively, the Schmid repeatability of conventional detectors under the same testing conditions (though without the max point cap of 500 points per scene) shown in Figure 5.14, indicates that the performance of conventional detectors (such as Fast and

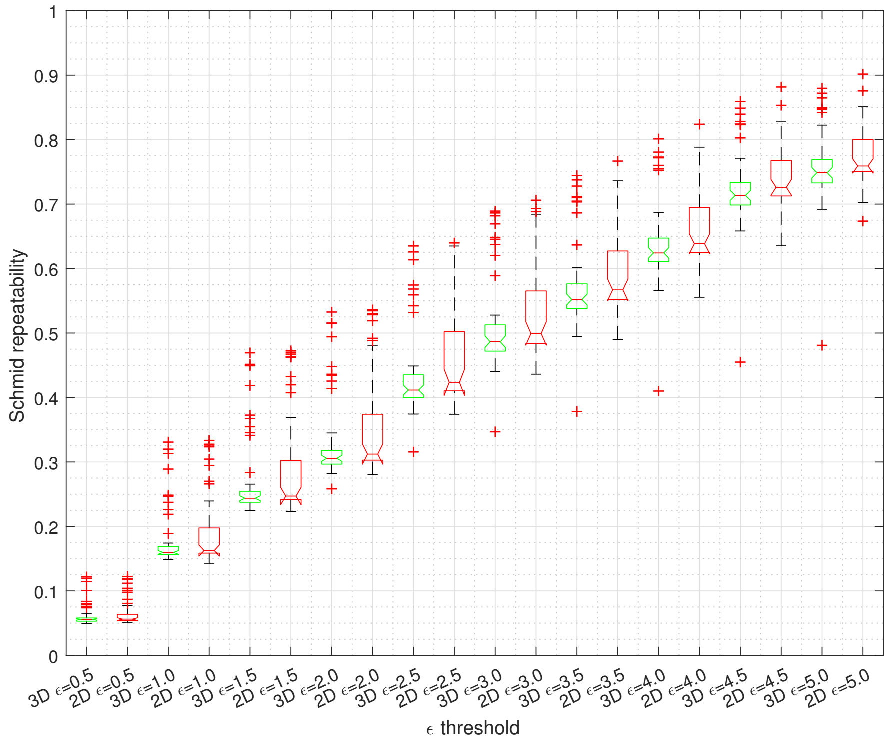


Figure 5.12: Analysis D: 2D(red box)/3D(green box) training with 12models and Schmid repeatability tested using asian dragon model. 50 training runs each.

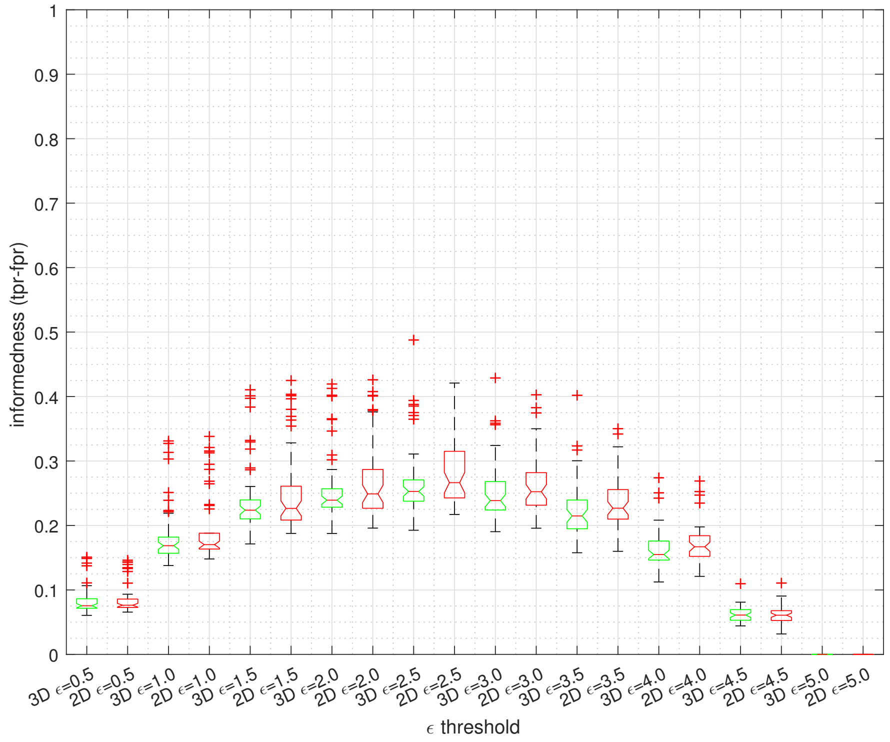


Figure 5.13: Analysis D: 2D(red box)/3D(green box) training with 12models and informedness tested using asian dragon model. 50 training runs each.

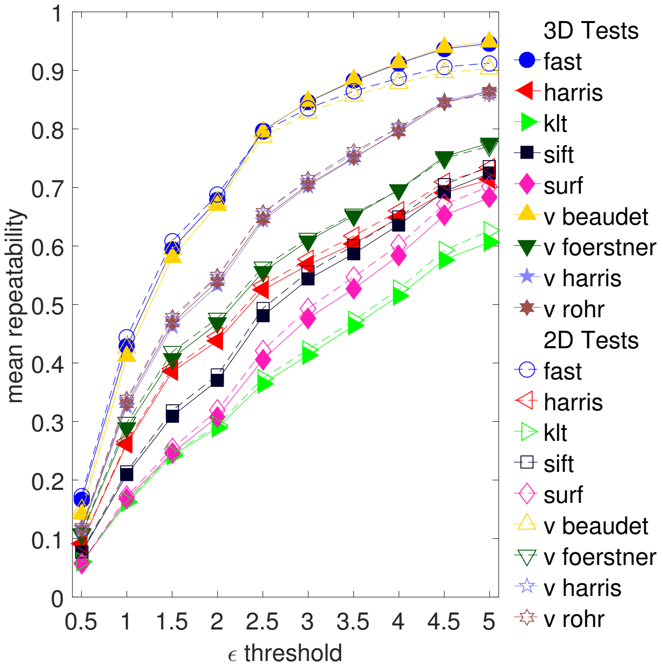


Figure 5.14: 2D (open-dashed) overlaid versus 3D (solid lines) Schmid repeatability of conventional detectors using the asian dragon model.

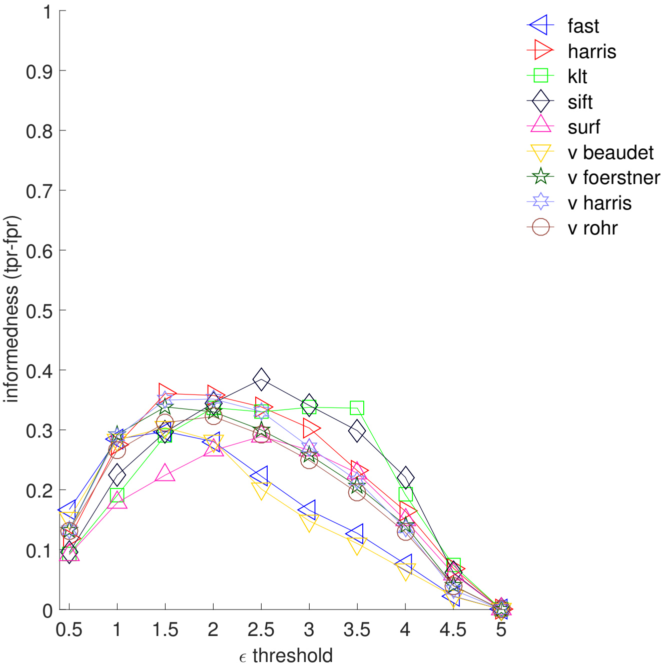


Figure 5.15: 2D/3D informedness of conventional detectors using asian dragon model.

Analysis D: 2D/3D train 12models, test: dragon (50 runs each)				
	Schmid repeatability		informedness	
ϵ	F>3.9381	p-value (α=0.05)	F>3.9381	p-value (α=0.05)
0.5	1.0996	0.29693	0.47345	0.49303
1.0	1.6412	0.20318	1.0078	0.3179
1.5	1.7414	0.19004	0.79044	0.37615
2.0	1.8444	0.17756	1.0605	0.30564
2.5	2.0699	0.15342	1.4678	0.22861
3.0	2.2338	0.13823	1.7865	0.18445
3.5	2.3845	0.12577	1.7823	0.18496
4.0	2.3563	0.128	0.92665	0.3381
4.5	2.4541	0.12044	0.028269	0.86682
5.0	2.4195	0.12306	NaN	NaN

Table 5.9: F- and p-value at each ϵ for analysis D.

Beaudet) shows better performance than the evolved classifiers in Figure 5.12. However, informedness, shown in Figure 5.13 shows that the outlier performance of some evolved classifiers at $\epsilon = 1.5$ is much better than the informedness performance of all conventional detectors in both 2D and 3D training. Figure 5.13 also shows that evolved classifiers demonstrate slightly better performance at $\epsilon = 2.0$ and $\epsilon = 2.5$ in a few isolated cases compared to conventional detectors in Figure 5.15.

The important distinction in this instance is that while the conventional detectors can have potentially higher candidate repeated points to boost informedness, the evolved classifiers, which are capped at 500 points maximum per scene, are forced to "do more with less", and are thus better optimised to find better features. The evolved classifiers' generalisation of the 12model training dataset also helps to avoid cases where overtraining is more likely and through regressive testing in GP, should be focusing on more robust features during training. The drawback to training with larger datasets is that it is a computational, and, by extension, time-consuming approach to training using larger datasets such as 12models.

5.5 Chapter Summary

This chapter utilised virtual spaces in conjunction with GP to train classifiers. Its goal was to explore the advantages of virtual spaces and determine any statistical differences in performance when training or testing with 2D/3D datasets, as well as the effects of training with the use of either 2D or 3D IPs. Additionally, the application of informedness in ways other than 2D/3D was explored, such as the performance difference when training with RGB and greyscale. Each of these goals was addressed through analyses A-D. Schmid repeatability was used exclusively during training, and for testing, both Schmid repeatability, and informedness was used during testing to measure classifier performance. To measure the performance differences between training and testing configurations, box plots, as well as the F-value and p-value were used. The best ϵ threshold was also empirically examined as part of informedness analysis.

The testing results from analysis A statistically verified that the use of RGB colour terminals performed better than the use of greyscale alone, even when the testing dataset did not extensively use colour textures. The best informedness trade-off was also statistically confirmed to be at $\epsilon = 2.0$. Conversely, when testing and training datasets were switched for analysis C, the testing of a single image created more variable results due to insufficient testing data. This demonstrated that testing with a single image did not produce enough generalisation to make reasonable determinations about testing configurations.

Analysis B established that testing 2D/3D results produced results that were statistically the same for classifier performance. This confirmed that the optimisation of classifiers operated similarly in both 2D and 3D training. It also indicated that outlier classifiers had the best informedness trade-off at $\epsilon = 2.0$ in both 2D and 3D training.

For analysis D, the use of 3D models for testing and training to compare training using 3D or 2D IPs did not produce statistically significant differences between training configurations. However, the use of 3D models did demonstrate that Schmid repeatability and informedness showed conflicting analysis when compared to conventional detector performance that were not restricted by the number of IPs. In Schmid repeatability, conventional detectors showed an advantage, however, for informedness, the evolved detectors had the advantage. This lack of concurrence showed that the inclusion of type II errors within the informedness metric can produce evaluations that diverge from Schmid's. Informedness also emipirically demonstrated that 3D training trade-off was highest at $\epsilon = 2.0$.

Note that medians probably give the best idea of performance, as the top outlier may be overtraining, or lucking into, features that work for the asian dragon model. In Chapter 6 we will reverse the training/testing datasets in the next chapter to explore how consistently performance of classifiers generalizes.

Chapter 6

Virtual Ground Truth, and Informedness Evaluation in GP: II

6.1 Introduction

The investigations in Chapter 5, used GP to optimise 2D classifiers in virtualised 3D spaces. As a result, 2D, 3D, RGB, and greyscale training was made possible with both 2D and 3D models, and acted as the basis for testing the potential advantages of virtual spaces, as well as building an empirical base to statistically determine which ϵ trade-off is best after training. The varying training configurations (5.3) utilised, however, in Chapter 5, made it rather difficult to isolate specific configuration parameters across more than one other configuration so that a more comparative assessment could be made, in all cases it could only test on variable, that being utilisation of 2D or 3D data to calculate repeatability for training. It has also been difficult to make direct comparisons between evolved and conventional detectors, due to the fact that color channels such as RGB are not utilised, nor are the detectors tested in this dissertation easily restricted to a maximum interest point threshold like the GP algorithm.

Additionally, based on the results of analysis B, the training/testing configuration potentially lacked generalisation and could have created the possibility that the singular asian dragon model test got lucky and produced skewed results. This "luckiness" is evident when examining the matched 2D/3D repeated points in Appendices C and D, where certain models performed better than others. The use of single, image-based datasets for analysis C also did not necessarily serve to properly test informedness. Conversely, analysis A and B produced results that were consistent with expectations where RGB would perform better optimisation overall than greyscale, but again, due to the unavailability of conventional detectors that used RGB for testing, and the fact that the 2D image (sower) did not take full advantage of aspects of the virtual scene, really means these configurations are not ideal. Training using 12models also highlighted the time-intensiveness of using larger datasets. This time-intensiveness in turn, affected the opportunity to be more statistically rigorous in comparing performance of different training/testing configurations. Other factors, like image resolution (earlier set at 300px) and light position were not taken into consideration. The absence of shadows in the scene may have caused issues that were not originally not taken into account due to the default light position projecting very few shadows. Because lighting up to this point had always been positioned in the center, or at the camera position, it is possible that it produced few shadows and limited the potential advantages for optimisation when using 2D and 3D data. Based on these considerations, a number of new testing approaches are considered in this chapter.

6.1.1 Chapter Goals

Bearing in mind the factors above, the focus of this chapter will be narrowed from the broader proof of concept tests that were less comparable from an analytical viewpoint, and more proof of concept, to a focus on creating conditions that rigorously test virtual scenes in a more directed manner. In particular, I will investigate whether 2D and 3D demonstrate any significant difference in performance under certain conditions that will allow for easier comparison with other scene configurations. Based on the results in previous chapters, these tests will continue to empirically examine the optimal ϵ threshold. So far, the results collected using the informedness metric has indicated $\epsilon = 1.5$ is not the optimal threshold in some cases, so assessing the optimal ϵ threshold will also form part of the analysis.

To summarise, the following research objectives will be carried out:

- Whether the reversal of training/testing datasets as seen in analysis D, with a single model for training and 12models for testing, will perform similarly to analysis D, without impacting performance.

- Whether raising the maximum number of points in a single scene from 500 to 2000 (which some conventional detectors return) is necessarily beneficial to performance, and how this could affect informedness substantially.
- Whether the use of shadows will make for better use of the virtual scene, in order to optimise evolved classifiers.
- Whether the resolution of the images processed by the classifiers/detectors will affect performance.

6.1.2 Chapter Organisation

This chapter's organisation will begin with a brief overview of methodology (Section 6.2) in the form of a breakdown of the testing and analysis being conducted. This will be followed by a discussion of the results (Section 6.3), which will be broken down into individual analysis sections to primarily assess 2D/3D performance. Lastly, Section 6.4 will conclude and summarise the results.

6.2 Experimental Setup

Due to the successful utilisation of GP with STEIPR in Chapter 5, and in an effort to only test GP algorithms and settings that have been demonstrated to work, there have been no substantial changes to the methodology used in Chapter 5. The GPLab settings, as seen in Table 5.1, remain unchanged from those in Chapter 5. The function and terminal sets used in Table 5.3.3 remain mostly unchanged, the only exception being that I_r, I_g and I_b will not be used in future tests even though they have been shown with 95% confidence, to provide better performance. The decision to not use I_r, I_g and I_b is mainly due to the fact that they make comparisons to conventional detectors more difficult. STEIPR's settings, as seen in Table 5.2, also remain mostly unchanged, though the "Viewport" (image resolution) and "Max points" setting will be mostly unchanged from chapter 5, except for certain runs. The transforms to the models, and other classifier settings such as non-maximal suppression, will remain unchanged. These changes to the settings are reflected in table 6.1 and Function and Terminal sets 6.1, 6.2 and 6.3.

$$F = \Big\{+, |+|, -, |-|, |I_{out}|, *, \div, I_{out}^2, \sqrt{I_{out}}, log_2(I_{out}), k \cdot I_{out}\Big\}, \tag{6.1}$$

$$\cup\Big\{\frac{\delta}{\delta u}G_{\sigma_D}, G_{\sigma=1}, G_{\sigma=2},\Big\}, \tag{6.2}$$

$$T = \Big\{I_{grey}, L_x, L_{xx}, L_{xy}, L_{yy}, L_y\Big\} \tag{6.3}$$

Parameters	Description and values
Max suppression nearest neighbor	2
Viewport	300x300, 600x600, 1000x1000
X rotation	-50° to +50° in 10° increments
Y rotation	-50° to +50° in 10° increments
Z rotation	0° to +180° in 10° increments, clockwise
X,Y scale	1.25, 1.5, 1.75, 2.0, 2.5, 3.0, 3.5, 4.0
Texture filtering	Anisotropic and Linear
Max points returned	500, 2000

Table 6.1: STEIPR settings.

6.2.1 Dataset Experimental Setup

To address the potential limitations of previous experiments, the decision was made to focus on a single dataset for training and subsequent testing. As a result of the positive results from analysis A (Section 5.4.1) the 12model dataset was chosen for testing, and the asian dragon model was chosen for training. The asian dragon model was chosen due to the fact that it is highly detailed, with a variety of 3D texturing, but also with numerous “protrusions” that make it a good candidate model for misclassification of repeated points on those types of features, especially edges with parts of the model in the foreground and background. It also has a non-symmetrical structure, which means that lighting positions from the top left and top right (relative to the camera position) would provide sufficiently different shadowing. The higher positioning of the lighting is an attempt to simulate a single source of light, similar to that which would be experienced outdoors, which would be a common situation, and produce shadowing that would be desirable to optimise for in real life.

The restriction to a singular model was advantageous, as it was much faster for training. However, it also raised concerns for potential overtraining and poor performance. Though overtraining was possible, the major priority was to test the advantages of virtual scenes in 2D and 3D situations in a variety of configurations and their subsequent analysis via informedness.

6.2.2 Training and Testing Configurations

In line with the goals of this chapter, Table 6.2 details training and testing configurations. To ensure the results could be proven with 95% confidence with a sufficiently large number of samples, a minimum of 50 training runs were performed for each configuration in 2D and in 3D. The exception being 200 runs each of a "baseline" configuration. Each virtual scene parameter was isolated into 6 main experiments, with each consisting of 2D, and 3D trained classifiers. These experiments separately focused on the reverse of analysis D in Chapter 5where the testing and training datasets were the opposite, a raising of the max points to 2000, light position, and image resolution, as shown in Table 6.3.

Training dataset	Testing dataset	Color	Training Runs	Light Position	Image Size	2D/3D	Max Points
dragon	12models	GS	200	center	300px	3D	500
dragon	12models	GS	200	center	300px	2D	500
dragon	12models	GS	50	center	300px	3D	2000
dragon	12models	GS	50	center	300px	2D	2000
dragon	12models	GS	50	top left	300px	3D	500
dragon	12models	GS	50	top left	300px	2D	500
dragon	12models	GS	50	top right	300px	3D	500
dragon	12models	GS	50	top right	300px	2D	500
dragon	12models	GS	50	center	600px	3D	500
dragon	12models	GS	50	center	600px	2D	500
dragon	12models	GS	50	center	1000px	3D	500
dragon	12models	GS	50	center	1000px	2D	500

Table 6.2: Testing and training configurations.

Analysis	Training dataset	Testing dataset	Color	Training runs	Light position	Image size	2D/3D	Max points
E	dragon dragon	12models 12models	GS GS	200 200	center center	300px 300px	3D 2D	500 500
F	dragon dragon	12models 12models	GS GS	50 50	center center	300px 300px	3D 2D	2000 2000
G	dragon dragon	12models 12models	GS GS	50 50	top left top left	300px 300px	3D 2D	500 500
H	dragon dragon	12models 12models	GS GS	50 50	top right top right	300px 300px	3D 2D	500 500
I	dragon dragon	12models 12models	GS GS	50 50	center center	600px 600px	3D 2D	500 500
J	dragon dragon	12models 12models	GS GS	50 50	center center	1000px 1000px	3D 2D	500 500

Table 6.3: Sorting of analysis of training runs.

6.2.3 Experimental Hardware, Training Load and Timeframe

The testing hardware was identical to what was used in Chapter 5, no additional changes to the testing conditions. The approximate time for each run, shown in Table 6.2 can be seen in Table 6.4. The main differences in run durations in this chapter was primarily due to the changes in the resolution of the image being processed by the candidate classifiers. The training time duration per run generally fluctuated by 20-30%.

Dataset	CPU hours per training run (avg.)	Total training runs	Total CPU hours	Time required with 12 concurrent processes	
				Hours	Days
dragon 300px	8	50	400	33	1.38
dragon 300px	8	200	1600	133	5.55
dragon 600px	13	50	640	53	2.2
dragon 1000px	27.5	50	1374	114	4.7

Table 6.4: Approximate training time required under ideal conditions.

6.3 Results and Discussion

The results of the training and testing of classifiers have been divided into separate sections according to Table 6.3. To help streamline the results and discussion, conventional detector performance figures are not shown in this chapter. Each configuration tested in Table 6.2 with evolved classifiers has also been tested with conventional detectors, and can be seen in Appendix B.

6.3.1 Analysis E

Figures 6.1 and 6.2 and Table 6.5 show the box plot performance of all runs for analysis E, as well as the F- and p-values of these runs' results for Schmid repeatability and informedness. The results can be compared to the conventional detector results in Figure 5.7, and 5.6, and show that, in terms of performance on the informedness metric, the Harris detector outperformed all of the evolved classifiers, with the maximum results for $\epsilon = 1.5$ and $\epsilon = 2.0$ being quite close to Harris. When using Schmid repeatability as a metric, however, evolved classifiers performed less well than conventional detectors. By comparing analysis E's results to those of analysis D in Section 5.4.4, it is apparent that the generalisation (resistance to overtraining) has not degraded performance (Figure 5.13) when compared to Figure 6.2, but that instead, boosted it significantly for informedness. The third quartile was skewed in the upper range in Figure 6.2 for analysis

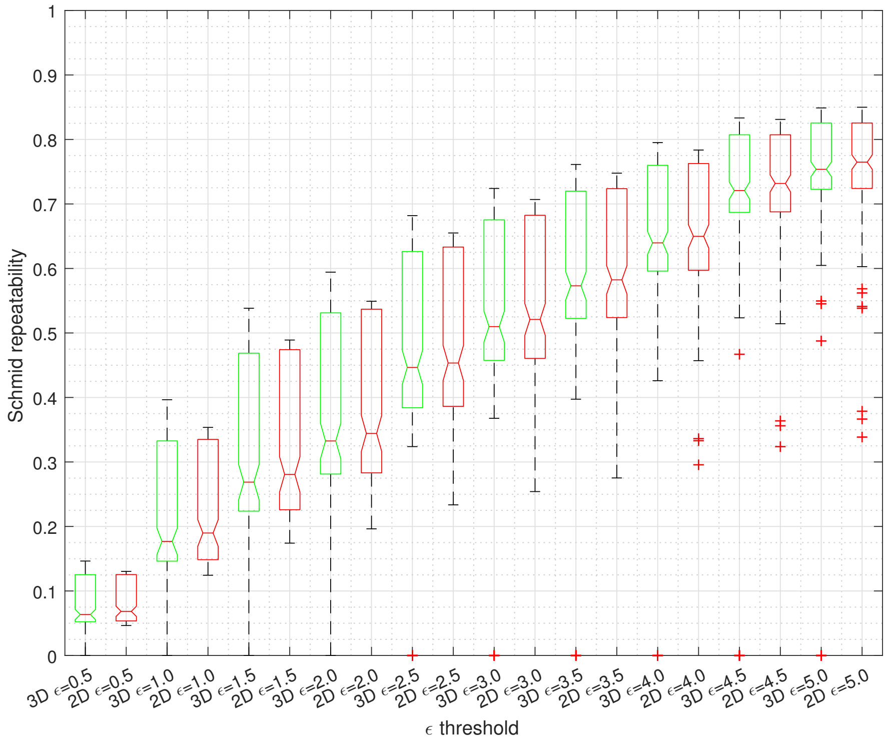


Figure 6.1: Analysis E: 2D(red box)/3D(green box) Schmid repeatability tested. 200 training runs each.

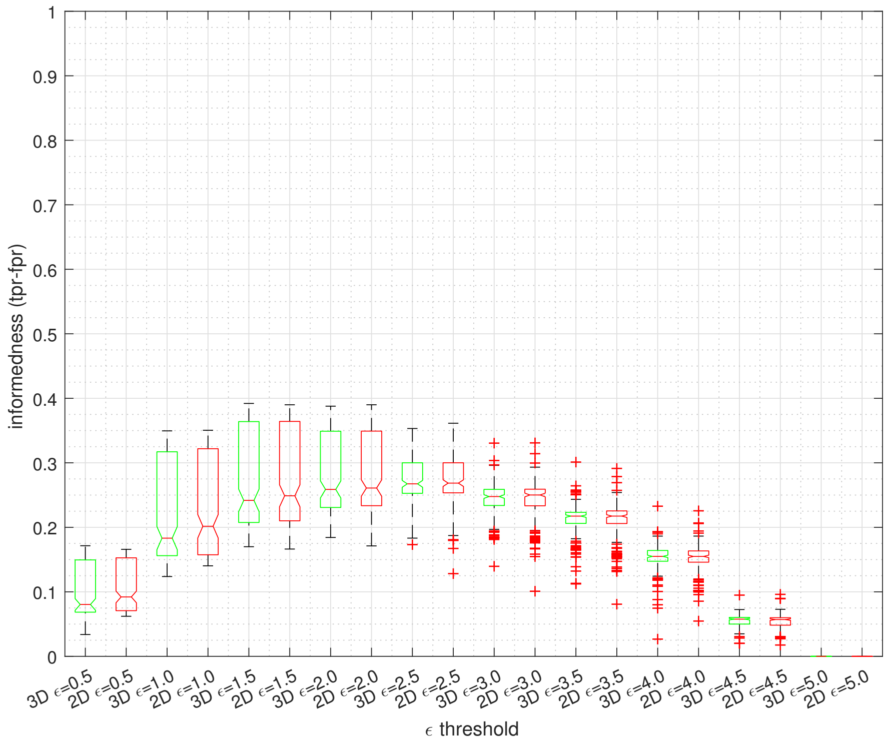


Figure 6.2: Analysis E: 2D(red box)/3D(green box) informedness tested. 200 training runs each.

Analysis E: 2D/3D 500 max points (200 runs each)				
	Schmid Repeatability		Informedness	
ϵ	F>3.8649	p-value (α=0.05)	F>3.8652	p-value (α=0.05)
0.5	0.8565	0.35528	0.76622	0.38192
1.0	0.85623	0.35536	0.47483	0.49118
1.5	0.89755	0.34401	0.14532	0.70325
2.0	0.95623	0.32873	0.023416	0.87846
2.5	1.0454	0.30719	0.21853	0.64042
3.0	1.143	0.28566	0.41403	0.52031
3.5	1.2212	0.26979	0.70256	0.40243
4.0	1.272	0.26007	0.48735	0.48552
4.5	1.3076	0.25352	0.45326	0.50118
5.0	1.2999	0.25492	NaN	NaN

Table 6.5: F and p-value at each ϵ for Analysis E.

E's informedness, compared to analysis D's informedness in figure5.13, which shows a much tighter interquartile range, and would likely imply that the single model testing only resulted in a few “lucky” classifiers getting the better results. Figure5.13 show more outliers on the training data that was tested with the asian dragon model, so would imply that when it comes to training and testing, it is better to utilise a larger dataset for testing, rather than to concentrate more training time on a larger dataset. The training experiments in analysis E, with the training/testing datasets switched, produced a trade-off that was arguably better than that in analysis D. Though the differences seemed to be only marginally significant at $\epsilon = 1.5$ and $\epsilon = 2.0$ when analysis D's informedness (Figure 5.13) and analysis E's (Figure 6.2) are compared in 2D and 3D training in each configuration, informedness performance is still similar, as well as better median performance of classifiers for analysis E (all be it marginally). Additionally, as seen in Table 5.5 and 6.4, training time with 12models for only 50 runs took 16.6 days, compared to 200 runs that only took 5.5 days. This difference in training time supports the decision to utilise a single model for training and a larger dataset for testing in order to produce better generalisation, as the effects of overtraining do not seem to be apparent in configurations that use the asian dragon model for training, and the 12model dataset for testing.

Focusing in on the 2D/3D performance of the training/testing configurations for analysis E specifically, there seems to be no significant 2D/3D performance difference at any ϵ threshold. With the number of runs being $N = 200$, it seemed to be definitive that, based on the configurations used, virtual spaces did not afford any performance benefit when it came to the utilisation of 3D interest points over 2D. It was also not apparent which ϵ threshold was optimal. $\epsilon = 1.5$ and $\epsilon = 2.0$ did not show any statistical difference in median performance, and Table 6.5 supports this result for both Schmid and informedness. Though the informedness median (Figure 6.2) at $\epsilon = 2.0$ is slightly higher and the maximums at both $\epsilon = 1.5$ and $\epsilon = 2.0$ were very similar, no conclusive differences could be found. However, with the exception of the Harris detector's performance in Figure 5.7, the maximum performance of evolved detectors did better than the rest of the conventional detectors when trained using 2D and 3D interest points.

Though these results are considered to be counter to the expected outcomes of the tests where we'd expect 3D trained classifiers to perform on average better than 2D, they establish that this configuration was much faster for testing in comparison to configurations such as that used in analysis D (5.4.4) with minimal performance degradation and better generalisation of results. This means that more variations of virtual scene configurations can be tested with better statistical confidence. However, with regards to the comparison of evolved detector performance to conventional detector performance, it is still somewhat difficult to isolate the potential bias of conventional detectors, due to their lack of a maximum point limit for each scene. To enable a more faithful comparison between evolved detector performance and conventional detector performance, evolved classifiers must be free to utilise more points in a scene.

6.3.2 Analysis F

In this analysis, the objective was to remove the possibility of the max points restriction impacting on performance and biasing conventional detectors with a potentially unfair performance advantage. Figure 6.3 and 6.4 and Table 6.6 show the box plot repeatability performance of all runs for analysis F, as well as the F- and p-values of these results for Schmid repeatability and informedness. The training performed in analysis F shared a similarity to E, with the exception that it limited the maximum interest points per scene to 2000 compared to the normal 500. This increased maximum allows us to assess whether the inclusion of more points affects classifier performance in a positive or a negative way. When comparing analysis E and F for Schmid repeatability in Figure 6.1 and Figure 6.3, there was a considerable statistical difference between 2D training and 3D training at every ϵ, and in all cases training with 2000 max points per scene is better. For informedness, 'when comparing results displayed in Figures 6.2 and 6.4, this was also reflected, but to a lesser extent, and only from $\epsilon = 0.5$ to $\epsilon = 2.0$.

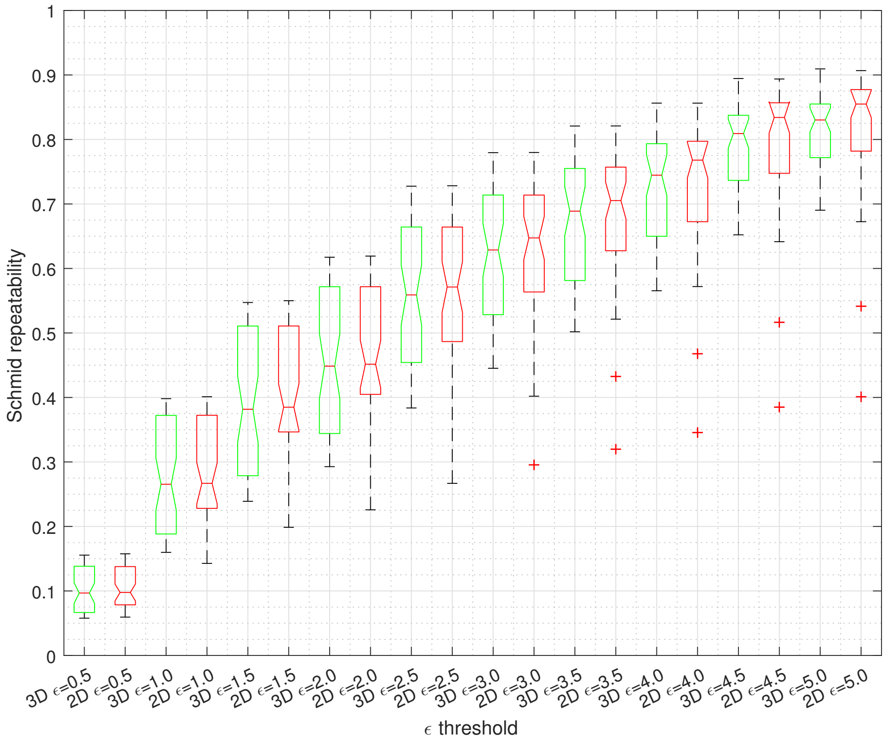


Figure 6.3: Analysis F: 2D(red box)/3D(green box) Schmid repeatability tested. 50 training runs each.

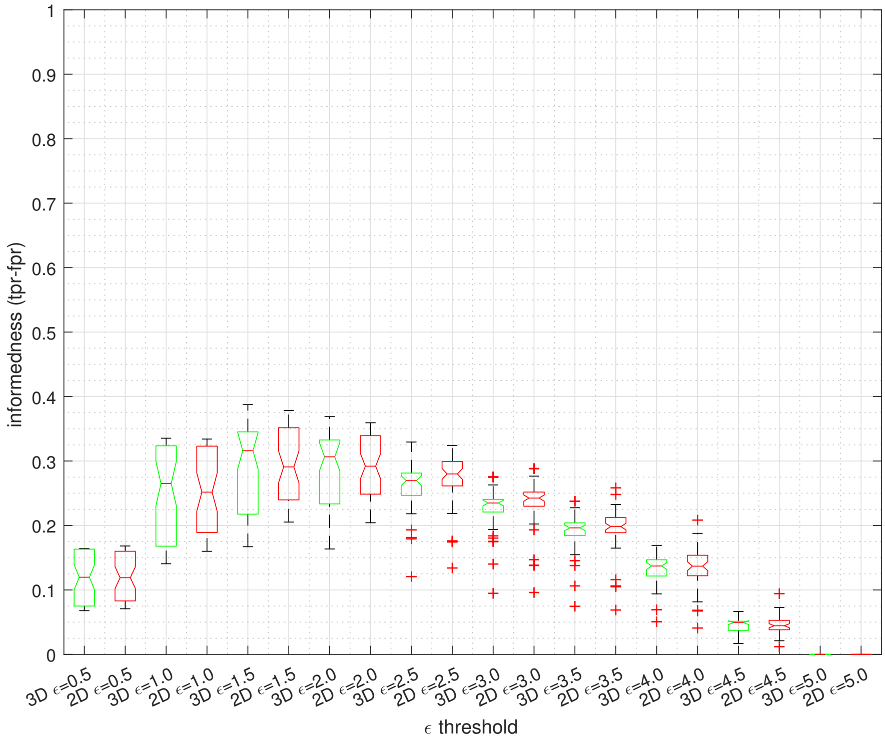


Figure 6.4: Analysis F: 2D(red box)/3D(green box) informedness tested. 50 training runs each.

Analysis F: 2D/3D 2kIP max points (50 runs each)				
	Schmid Repeatability		Informedness	
ϵ	F>3.9381	p-value (α=0.05)	F>3.9381	p-value (α=0.05)
0.5	0.093752	0.76011	0.0485	0.82615
1.0	0.083875	0.77273	0.074199	0.78589
1.5	0.1212	0.72848	0.37565	0.54136
2.0	0.15228	0.69721	0.64245	0.42476
2.5	0.22722	0.63465	1.5427	0.21718
3.0	0.26225	0.60973	1.1148	0.29363
3.5	0.27112	0.60376	0.52481	0.47052
4.0	0.23545	0.62859	0.1972	0.65797
4.5	0.15145	0.698	0.060831	0.8057
5.0	0.08655	0.76923	NaN	NaN

Table 6.6: F and p-value at each ϵ for Analysis F.

However, the maximums showed that the best candidates of these populations were very close to each other between analysis E and F at each ϵ. From a median standpoint however, a higher maximum point did translate to improved median performance in both 2D and 3D training.

Increasing the max point limit did not however result in a substantial performance advantage over conventional detectors as stated earlier, as the maximums are generally the same. Only the interquartile range, and median shows an increase. The 2D/3D performance comparison when using 2000 maximum points for training and testing, shown in Table 6.6 also does not indicate any statistically significant difference. When comparing the informedness performance of analysis E tested with 12models (Figure 6.2) to analysis D with the dragon model (5.13) and analysis F tested with 12models (Figure 6.4) to analysis D with the dragon model (Figure 5.13), the results are largely the same as analysis E.

The best ϵ trade-off in Figure 6.4 is also ambiguous at the $\epsilon = 1.5$ and $\epsilon = 2.0$ thresholds though. However, in the case of 3D training, $\epsilon = 1.5$ is arguably better, as $\epsilon = 1.5$ does perform better than 3D training at $\epsilon = 1.0$ with a the 95% confidence level, and overall has a higher maximum range. Though this would be a speculative interpretation that can't be statistically proven with sufficiently high confidence.

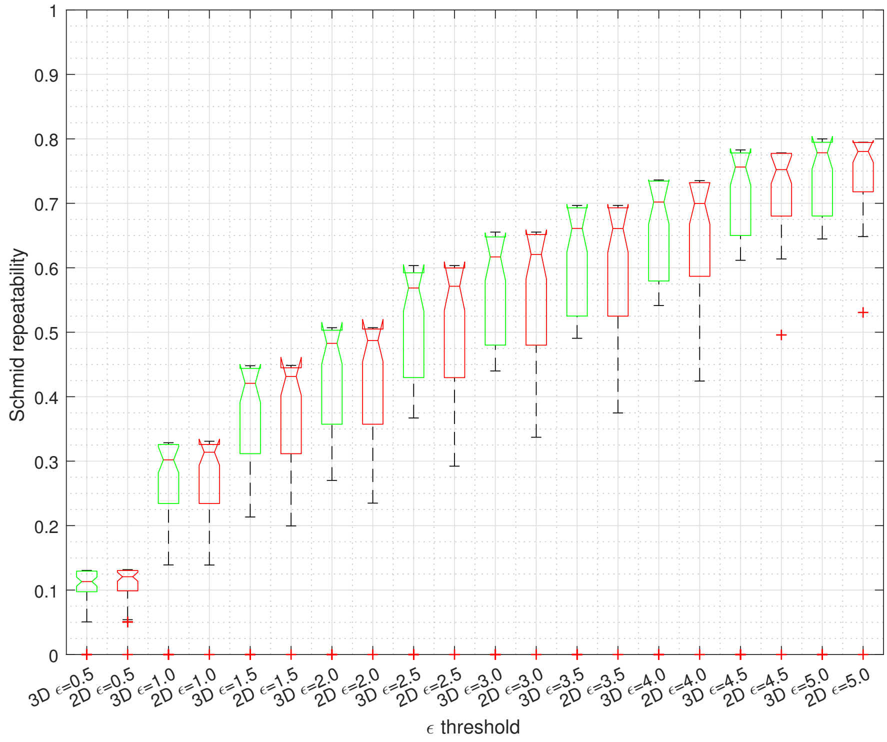


Figure 6.5: Analysis G (light position top left): 2D(red box)/3D(green box) Schmid repeatability tested. 50 training runs each.

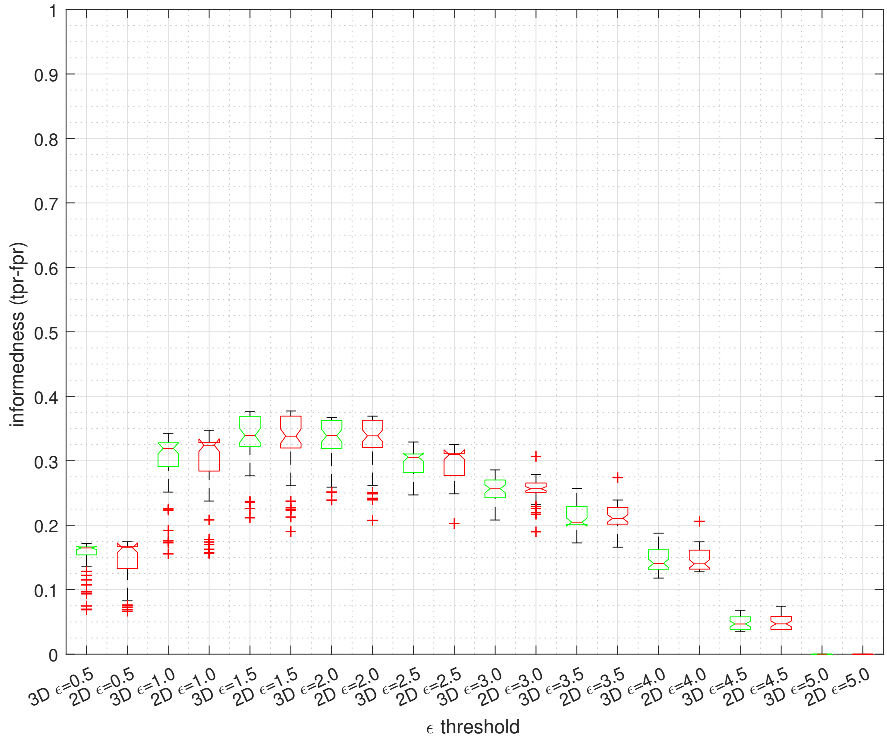


Figure 6.6: Analysis G (light position top left): 2D(red box)/3D(green box) informedness tested. 50 training runs each.

6.3.3 Analysis G

In this analysis, as well as in Section 6.3.4, the objective was to assess the 2D/3D training advantages of utilising different lighting conditions, and to observe how differing lighting positioning affected Schmid repeatability, and informedness performance. In this instance, the light was in the upper left relative to the position of the camera. Figures 6.5 and 6.6 and Table 6.7 show the box plot performance of all runs for analysis G, as well as the F- and p-values of these results for Schmid repeatability and informedness. Figure 6.5 it indicates that based on Schmid repeatability, median performance of classifiers was very tight towards the maximum in the upper third quartile, and skewed in the lower second quartile. For informedness (Figure 6.6) a similar skew did not seem evident. In both figures, there was no clear difference in median performance for either 2D or 3D training, and the most optimal ϵ trade-off was unclear, as $\epsilon = 1.5$ and $\epsilon = 2.0$ shared similar maximum ranges. The lack of difference in 2D/3D training performance was further supported by Table 6.3.1 which indicated no statistical differences at any ϵ thresholds between 2D and 3D training.

Analysis G: 2D/3D light top left (50 runs each)				
	Schmid Repeatability		Informedness	
ϵ	F>3.9381	p-value (α=0.05)	F>3.9423	p-value (α=0.05)
0.5	0.14678	0.70247	0.16346	0.68691
1.0	0.1169	0.73316	0.19905	0.65651
1.5	0.13137	0.7178	0.1138	0.73661
2.0	0.17203	0.67922	0.06956	0.79256
2.5	0.24901	0.61889	0.00026994	0.98693
3.0	0.33256	0.56548	0.10384	0.74799
3.5	0.39476	0.53127	0.232	0.63116
4.0	0.48406	0.48824	0.062999	0.80237
4.5	0.6177	0.4338	0.0067671	0.93461
5.0	0.67848	0.41211	NaN	NaN

Table 6.7: F and p-value at each ϵ for Analysis G.

6.3.4 Analysis H

Similarly to Section 6.3.3, this configuration, that use a slightly diffierent light position, aims to test the effects of 2D and 3D training with the light in the upper right, relative to the position of the camera. Figure 6.7, 6.8 and Table 6.8 show the box plot performance of all runs for analysis H, as well as the F- and p-values of these results for Schmid repeatability and informedness. As noted earlier, the non-symmetry of the training model permit a range of new light-induced features based on a change in lighting positions. Appendix A.3 shows the different reference transforms used in training depending on whether the lighting is situated in the top left, or top right.

Again, we observed no statistical difference in 2D and 3D training in either Schmid repeatability, or informedness, and the trade-off seems to peak at $\epsilon = 1.5$ but the best performance trade-off according to informedness (6.8) was inconclusive, as $\epsilon = 2.0$ is also in a similar range. Table 6.8 also can't establish with any confidence, that there is a statistical difference between 2D and 3D training. A very strong median difference can be seen between analysis G's Schmid repeatability (Figure 6.5) and that of analysis H's (Figure 6.7), which shows an upper skew in the third quartile for analysis G, and lower skew in the second quartile for analysis H. The light position had a very large effect on the median performance, such that though the maxima between analysis G and H, at least for $\epsilon = 1.5$ and $\epsilon = 2.0$, were much the same in both cases, but the median were significantly different. The light, positioned in the top right of the visual field relative to the camera position, has skewed Schmid results downwards, while the opposite is true for the light in the top right.

Informedness, on the other hand, resulted in a much tighter interquartile range for analysis G (Figure 6.6), and a much wider interquartile range for analysis H's informedness in Figure 6.8. From $\epsilon = 0.5$ up to $\epsilon = 2.5$, median performance was statistically divergent in both 2D and 3D training between analysis G (light top left)(Figure 6.6) and H (light top right)(Figure 6.8), and reflected that the light positioning has substantially impacted informedness performance, similarly to Schmid. Informedness, however, did not have skewed data, and instead had a contraction of the interquartile range. Based on the difference noted previously it indicates training, irrespective of whether it be using 2D or 3D data, favors lighting in the top corner, rather than the right right. Both Schmid repeatability and informedness reflect this performance improvement such that they agree with each other, based on light positioning, just in slightly different ways. Median informedness performance (shown in Figure 6.2) was also better at $\epsilon = 1.5$ and $\epsilon = 2.0$ when compared to performance in analysis F, and analysis E from $\epsilon = 0.5$ to $\epsilon = 2.5$, though the maximum range of E classifiers was slightly higher.

So far in this chapter, Schmid and informedness has largely produced similar results regarding median classifier performance between ϵ thresholds, as well as in

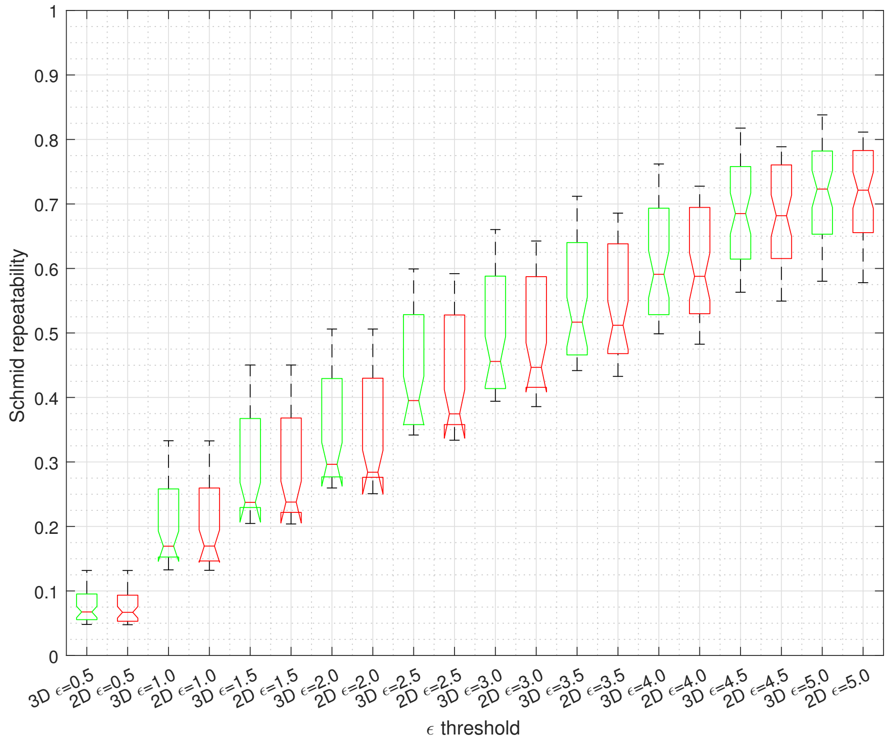


Figure 6.7: Analysis H (light position top right): 2D(red box)/3D(green box) Schmid repeatability tested. 50 training runs each.

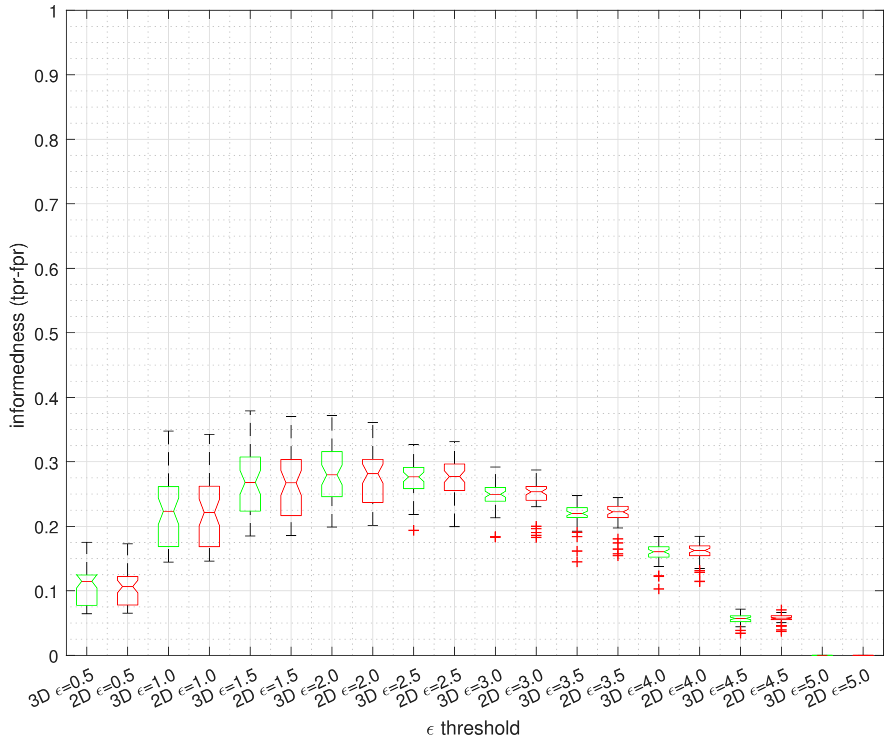


Figure 6.8: Analysis H (light position top right): 2D(red box)/3D(green box) informedness tested. 50 training runs each.

Analysis H: 2D/3D light top right (50 runs each)				
	Schmid Repeatability		Informedness	
ϵ	F>3.9381	p-value (α=0.05)	F>3.9381	p-value (α=0.05)
0.5	0.025156	0.87431	0.032921	0.8564
1.0	0.037349	0.84716	0.033088	0.85604
1.5	0.050589	0.82251	0.041276	0.83943
2.0	0.069447	0.7927	0.022743	0.88044
2.5	0.10161	0.75058	0.0061816	0.93749
3.0	0.11622	0.7339	0.019355	0.88964
3.5	0.13105	0.71812	0.056417	0.81275
4.0	0.14073	0.70836	0.26828	0.60566
4.5	0.13496	0.71414	0.22732	0.63458
5.0	0.11847	0.73144	NaN	NaN

Table 6.8: F and p-value at each ϵ for Analysis H.

analysis A (Section 5.4.1) when analysing color and greyscale terminals, and has no given conflicting analyses. This implies that at the very least as an analytical tool, informedness could serve well as a replacement to Schmid, due to its advantage in demonstrating the best performance trade-offs (which Schmid is not capable of), and can converge on the median with fewer runs, which is shown by the tighter interquartile ranges in many cases.

6.3.5 Analysis I

In order to explore other factors that may influence virtual space performance, the resolution of images has been investigated in this dissertation. The advantages of investigating the resolution of images did not necessarily seem apparent, as it could substantially increase training time due to the classifiers needing to process a larger area, and there were concerns that the dispersal of the same numbers of points over a far larger area would not yield any new insights or create opportunities to better utilise the search space. Conversely, the use of larger resolutions presented the potential for more robust feature detection, simply due to more complex features being available to optimise from. The use of 300x300 pixel viewports also presented the problem that the system was under-utilising the very highly-detailed models being used, and, as such, would result in fewer robust features. Figures 6.9 and 6.10 and Table 6.9 show the box plot performance of all runs for analysis I, as well as the F- and p-values of these results for Schmid repeatability and informedness. Looking at Schmid repeatability in Figure 6.9, the 3D training shows the interquartile range was skewed upwards, whereas this

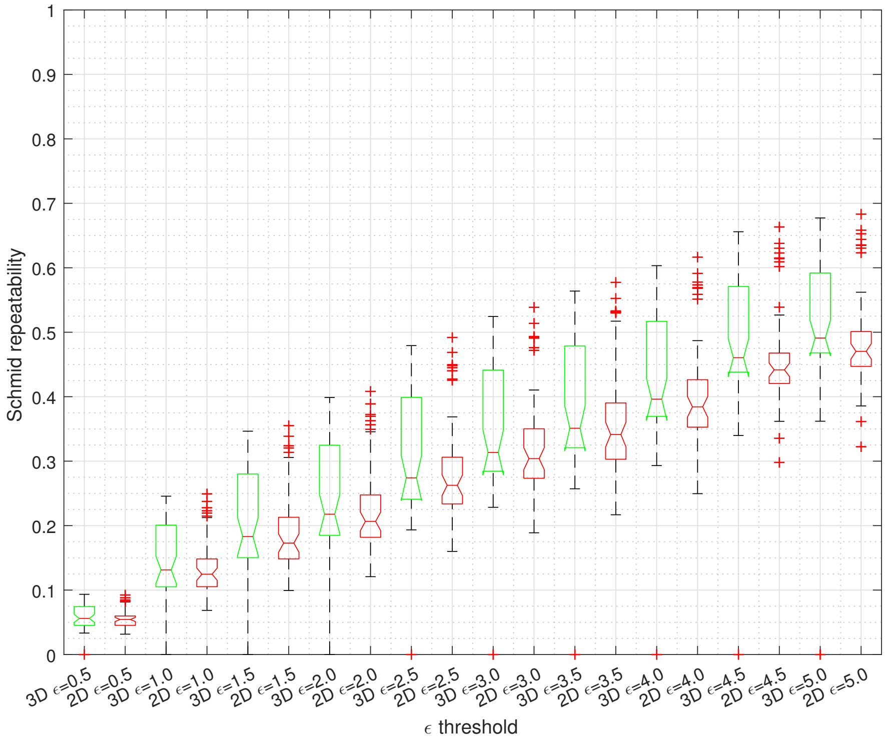


Figure 6.9: Analysis I (600px): 2D(red box)/3D(green box)d Schmid repeatability tested. 50 training runs each.

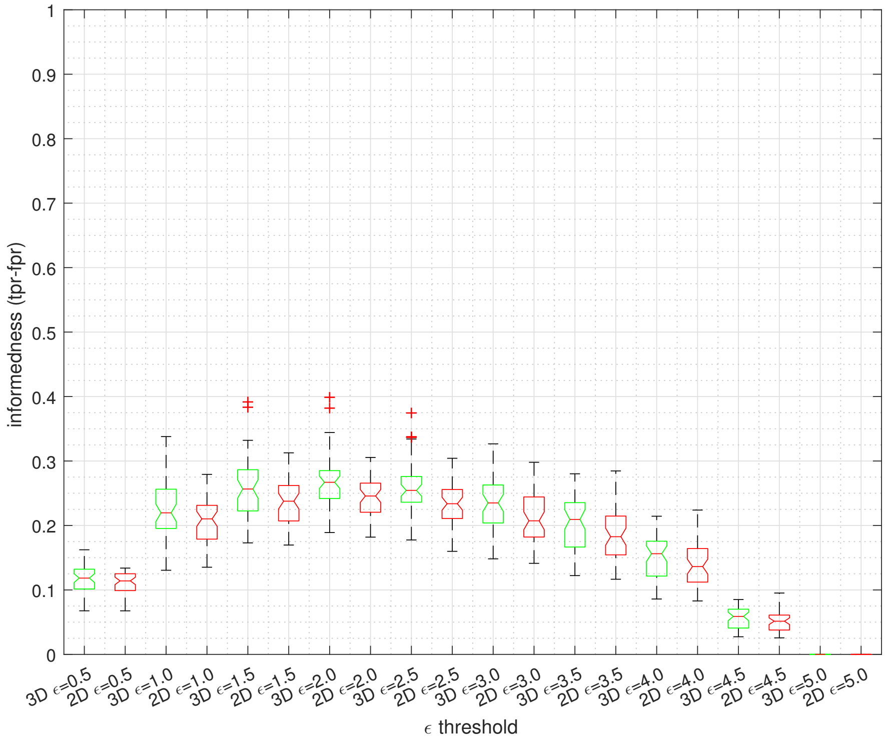


Figure 6.10: Analysis I (600px): 2D(red box)/3D(green box) informedness tested. 50 training runs each.

Analysis I: 2D/3D 600px (50 runs each)				
	Schmid Repeatability		Informedness	
ϵ	F>3.9381	p-value (α=0.05)	F>3.9381	p-value (α=0.05)
0.5	1.7585	0.1879	1.162	0.28373
1.0	1.7378	0.19049	4.4351	0.03779
1.5	1.5974	0.20926	9.3593	0.0028685
2.0	1.5583	0.21489	12.066	0.00076912
2.5	1.4179	0.23662	11.154	0.001191
3.0	1.4039	0.23893	7.7657	0.0064062
3.5	1.4242	0.2356	5.2174	0.024542
4.0	1.4368	0.23355	3.7137	0.056893
4.5	1.4888	0.22534	2.6154	0.10908
5.0	1.467	0.22873	NaN	NaN

Table 6.9: F and p-value at each ϵ for Analysis I.

was not the case in 2D training. However, 2D and 3D training does not show any statistical differentiation between them. The maximum range was higher in 3D; however, the 2D box plots still had similarly performing outliers at that level to defer the formation of any conclusions regarding a definitive performance advantage. Informedness, on the other hand, (shown in Figure 6.10), did show 3D training has statistical difference in median performance at $\epsilon = 1.5$ and $\epsilon = 2.0$ (at its strongest) when compared to 2D at those same thresholds. Table 6.9 also supported this with the F-value for $\epsilon = 1.0$, $\epsilon = 1.5$ and $\epsilon = 2.0$ greater than the F-critical 3.9381 (with a significance level of 0.05), as well as the p-value at each of those ϵ thresholds supporting this, with values below the 0.05 significance level. Though this was not necessarily apparent in the informedness box plot due to its close margins, the F- and p-values strongly supported a divergence in 2D and 3D performance up to $\epsilon = 3.5$. However, such high thresholds in normal circumstances may not be very beneficial, as the best performance trade-off peaks at around $\epsilon = 2.0$. The p- and F-values also supported $\epsilon = 2.0$ as the best trade-off with the strongest results shown in Table 6.9. With $N = 50$ training runs, and the 12model testing dataset affording better generalisation, these results indicated that training with the use of 3D data for optimisation is preferable to optimisation with 2D alone when using this specific configuration.

Most importantly, the performance difference between 2D and 3D, was not reflected in the Schmid repeatability statistical analysis in Table 6.9, nor in Figure 6.9. Schmid repeatability's inability to notice this performance difference would likely indicate that where informedness was capable of detecting/accounting for type II errors, the Schmid algorithm failed to detect/account for such errors.

Another important factor was that this better performance was being optimised for passively by the GP algorithm, as the GP algorithm (and by extension, the classifiers it was optimising), had no knowledge of the scene beyond the image the STEIPR program provided the classifier for processing, and had no information regarding the preselecting of closest points based on 2D or 3D distances between interest points, as was initially presented in Chapter 4. This form of optimisation, simply through discriminating closeness based on Euclidean distance rather than 2D distances, presents a strong argument for the testing of 2D keypoints with virtualised ground truths. It also supports the use of informedness as an analytical tool that can identify these types of performance improvements to which Schmid is blind.

6.3.6 Analysis J

In addition to the 600px image sizes, 1000px image sizes were also tested. Figures 6.11 and 6.12 and Table 6.10 show the box plot performance of all runs for analysis H, as well as the F- and p-values of these results for Schmid repeatability and informedness. The use of 1000px images did not afford the same improvement here as it did in analysis I, and by comparing the median ϵ for both Schmid repeatability and informedness, probably did poorest out of all the tests done in this chapter. As Table 6.4 also highlights, training of classifiers that used 1000px images took approximately 50% longer to train compared to the 600px training runs, and 70% longer compared to a 300px training run. It also did not produce any statistically significant difference in training with 3D over 2D at any ϵ threshold, which was also supported by the results in Table 6.10.

This raises speculations over why this may have been the case, and shows that when compared to conventional detectors shown in Figures B.11 and B.12, that even the best evolved detectors considerably worse, when conventional detectors seem to perform far better with larger image sizes, SURF being one of the few exceptions where the evolved classifiers perform better in Schmid repeatability and informedness. One possibility could be that the datasets did not have sufficient detail to produce robust points, such as smoother surfaces, as the model could no longer provide complex detail at higher resolutions (though this line of thinking would conflict with the fact that conventional detectors still performed very well). However, the fact that the scene's max points were set to 500 could have limited the numbers of candidate repeated points. The initial concern regarding model detail by moving to 600px may be becoming more likely with these much higher

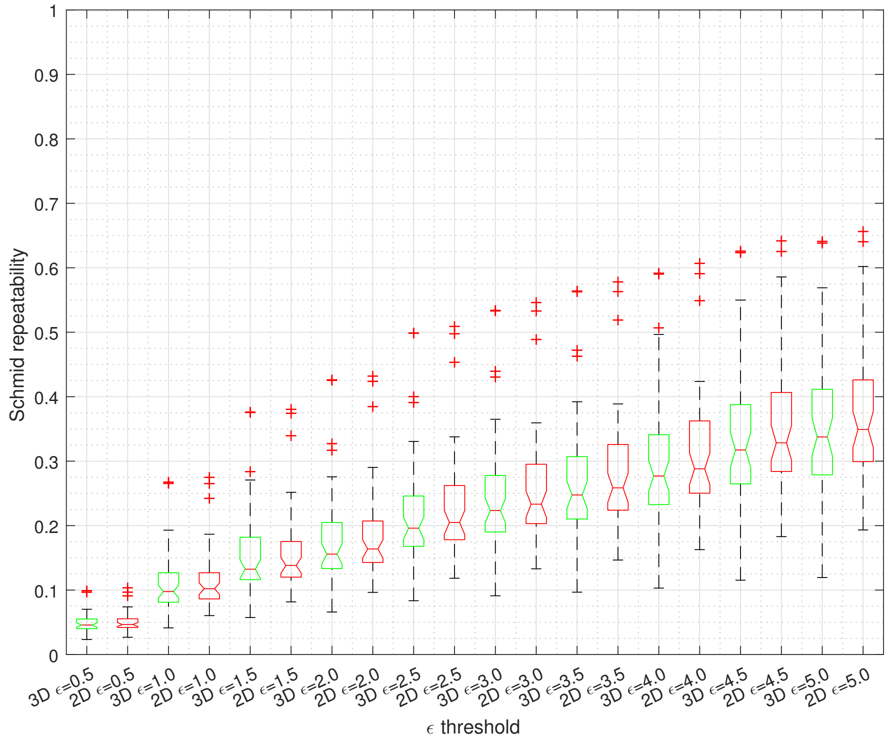


Figure 6.11: Analysis J (1000px): 2D(red box)/3D(green box)Schmid repeatability tested. 50 training runs each.

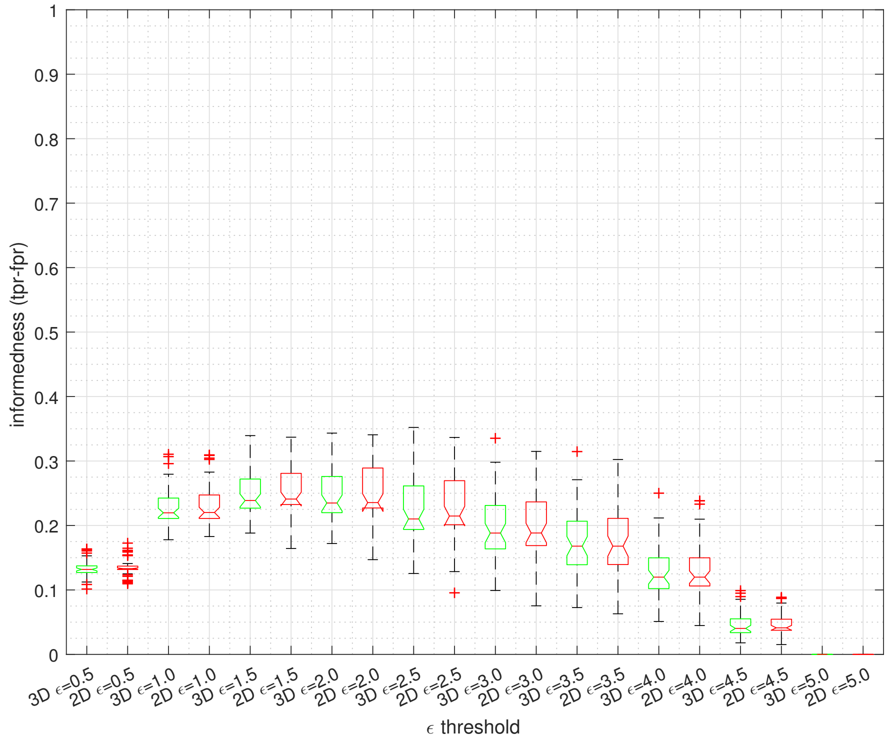


Figure 6.12: Analysis J (1000px): 2D(red box)/3D(green box) informedness tested. 50 training runs each.

Analysis J: 2D/3D 1000px (50 runs each)				
	Schmid Repeatability		Informedness	
ϵ	F>3.9381	p-value (α=0.05)	F>3.9381	p-value (α=0.05)
0.5	0.51109	0.47637	0.074702	0.78519
1.0	0.2082	0.64919	0.39511	0.53109
1.5	0.13785	0.71123	0.4528	0.50259
2.0	0.11374	0.73664	0.19356	0.66094
2.5	0.08855	0.76666	0.2929	0.58959
3.0	0.095747	0.75765	0.18256	0.67012
3.5	0.10989	0.74097	0.15389	0.69569
4.0	0.13568	0.7134	0.15726	0.69255
4.5	0.16688	0.68379	0.00012865	0.99097
5.0	0.17469	0.67689	NaN	NaN

Table 6.10: F and p-value at each ϵ for Analysis J.

resolutions but also could have been compounded by the max point cap. When compared to conventional detectors that were tested using the same configuration, Figure C.11 illustrates that the maximum numbers of points exceed 9000 in some scenes and the informedness results in Figure B.12 demonstrate a marked improvement in informedness performance as a result.

6.4 Chapter Summary

The goal of this chapter was to investigate how certain elements of the testing environment affected the performance of 2D and 3D training via GP. Various factors such as training/testing datasets, limiting the number of points in each scene, lighting positions, and how performance was affected by the resolution of the scene, have been considered. Conventional Schmid repeatability and informedness were used to not only compare performance between 2D and 3D, but to measure where the best performance trade-off occurred and how this varied between analyses.

From the results of these tests, analysis E found that a single model for training was sufficiently complex to provide performance results that were competitive compared to most conventional detectors, as well as significantly faster for training purposes. Simultaneously, a larger testing dataset provided sufficient generalisation without substantially detracting from performance. Analysis F demonstrated that the use of a higher max point limit did provide an improvement in median performance that was statistically significant compared to the performance presented in analysis E's results. Though the maximum ranges did not differ both Schmid and informedness confirmed a performance improvement. Analyses G and H both revealed that the non-symmetrical datasets affected performance substantially when lighting positions changed. However, in the case of Schmid, non-symmetrical datasets caused the data to skew upward/downwards, whereas in the case of informedness, the interquartile ranges were wider/narrower. It also demonstrated that analysis H had the best median informedness performance between E, F and H as well as the tightest interquartile range, though analysis F had the highest maxima at $\epsilon = 1.5$ and $\epsilon = 2.0$ when compared via informedness.

Analysis I showed the most promise for demonstrating the advantages of training with 3D data over 2D, with 3D median performance the F- and p-values showing a statistical significant diffierence in median performance at $\epsilon = 1.0$ $\epsilon = 1.5$ and $\epsilon = 2.0$. This was only supported by informedness however, and Schmid showed no indication of a performance improvement. Informedness also demonstrated that the best ϵ performance tradeoff was most optimal at $\epsilon = 2.0$. Analysis J's use of much larger resolution of 1000px did not afford any substantial improvement, perhaps due to insufficient detail of the dataset which could have affected training at higher resolutions. The limitation of the 500 max points per scene could have also played a factor, which restricted the numbers of interest point candidates that could be repeated and analysed.

Chapter 7

Conclusions and Future Work

The primary goal of this thesis was to investigate the use of virtual spaces for evaluation of 2D-based detector performance, with a particular focus on existing repeatability methodologies and the pursuit of alternative evaluation approaches that could be afforded by virtual spaces. This goal was achieved by developing a virtual scene framework that could map 2D keypoints to features on 3D-scanned models, such that it could track features with a high degree of accuracy and emulate already-existing repeatability evaluation based on 2D or 3D information within the scene. As well as this, based on the informedness measure, an alternative evaluation of repeatability performance was adapted with the additional information that 3D scenes afforded. These new and existing methods, which have been adapted to virtual spaces, were tested using conventional, well known and used detectors, as well as with GP to verify the performance of classifiers that were optimised with virtual spaces under various conditions. The virtual spaces themselves utilised a variety of datasets, varying from singular 2D images, such as the well-known sower image and the BSDS500 dataset, to highly detailed 3D-scanned models (with examples including the Stanford dragon/bunny/asian aragon models), and various others under a variety of affine transforms, lighting conditions, noise conditions, GP training settings, resolutions, and RGB/greyscale channels.

The rest of the chapter summarises the research goals achieved by this thesis and the main conclusions from Chapters 3- 6, and discusses what can be built on for future research.

7.1 Research Goals Achieved

This thesis achieved the six research objectives outlined in Section 1.3.

1. The thesis developed a virtualised 3D space to test existing computer vision algorithm performance with 3D models in Chapter 3, with the necessary functionality required to duplicate existing performance evaluation of interest points. A means of emulating performance evaluation of 2D keypoints originally described by Schmid in a virtualised 3D space with 2D images and 3D models concurrently was developed. Creating a better system for the establishment of ground truth allowed for a reliable performance evaluation that could discriminate performance in ways that were not possible with 2D images. The creation of this system also granted better discrimination of feature classification in the virtualised 3D environment in a way that more accurately reflected real world environments.

2. This thesis utilised virtual spaces to emulate testing conditions that are normally used to test already-existing interest point detectors (Chapter 3). Schmid's performance evaluation was adapted for use in virtual spaces such that 2D keypoints could be evaluated across multiple ϵ thresholds in 3D. This permitted the evaluation of 2D keypoints of detected 3D features in environments that had not been normally tested.

3. This thesis developed other analytic strategies (based on the 3D ground truth) that are able to provide better analysis of performance than existing performance metrics described in Chapter 4. With a highly accurate and reliable ground truth afforded by the virtual space, other evaluation methodologies based on ROC analysis were demonstrated that incorporate the detection of type I and type II errors, as compared to Schmid's approach to classification and evaluation, which only utilises type I. Informedness, a well researched, statistically reliable, but little-utilised form of evaluation, was able to normalise ROC data in a way that was more useful for the application domain (evaluation of interest points in 2D/3D scenes with a reliable ground truth), to evaluate overal interest point performance at each ϵ threshold, in a manner similar to Schmid repeatability. Furthermore, informedness empirically demonstrated its ability to measure the best ϵ performance trade-off which was not possible with Schmid repeatability. In

this way, it was demonstrated that informedness can be used to make direct comparisons between detectors within a controlled testing environment.

4. This thesis integrated the 3D testing environment with existing GP training algorithms to demonstrate that it is capable of performance evaluation and optimisation of interest point detectors unassisted in Chapter 5. The optimisation of classifiers utilised existing GP algorithms, and their performance was tested using Schmid repeatability and informedness. Schmid repeatability and informedness was compared in 2D and 3D interest point repeatability, with 2D images and 3D models which verified optimisation in 3D was viable. After training, informedness was proven to be reliable at discriminating performance of classifiers similarly to Schmid with a similar level of precision to that afforded by Schmid repeatability, while also being able to perform analysis of optimisation beyond 2D and 3D, such as optimisation based on colour and greyscale. Informedness also empirically supported the argument that $\epsilon = 1.5$ was not necessarily ideal as a performance tradeoff.

5. This thesis demonstrated that evolved detectors can perform at a similar level as most conventional detectors, tested with far fewer points neccessary (Chapter 6), but also demonstrated that at least one type of implementation of Harris has excellent performance and that this implementation still performs well in 3D environments as opposed to other conventional detectors, some of which are much more recent. Experiments statistically support that training classifiers with higher numbers of points per scene also afforded better performance ('in contrast to existing research, which recommended lower numbers of points per scene). “Fast” also demonstrated better performance than evolved classifiers, but it is arguable that this was only due to bias as a result of higher numbers of points.

6. This thesis demonstrated that the utilisation of 3D depth can afford a measurable improvement in detector performance when compared to the absence of depth (Chapter 6). The utilisation of training with 600px images of the virtual scene demonstrated the median classifier performance was better when trained in 3D compared to 2D, and also that $\epsilon = 2.0$ provided a better performance trade-off compared to $\epsilon = 1.5$. This performance improvement was only detected by informedness however, and not by Schmid. This demonstrated that the integration of additional data due to the virtual scene's ground truth helped account for type II errors that were not incorporated by Schmid's approach.

7.2 Main Conclusions

The main conclusions based on the six research objectives in Section 1.3 are discussed in this section.

7.2.1 Virtual Ground Truth

Chapter 3 investigated the utilisation of virtual spaces with 2D keypoint detectors by emulating already-existing performance algorithms developed by Schmid. The goal of adapting virtual spaces to test 2D keypoints was successfully achieved by modifying the implementation of Schmid's originally-described algorithm to duplicate 2D implementation of repeatability while enabling 3D scenes to be seamlessly integrated. The adapted performance algorithms developed by Schmid were tested using 2D images and 3D models under a variety of well-accepted scene transforms in the form of affine transforms on the image/model. The performance of each image/model was measured in a simplified, highly-controlled environment. To do this, Schmid's original use of a homography was replaced with the use of inverse transforms of 2D keypoints, which were inverse mapped to Euclidean space co-ordinates. Through the replacement of the homography with an alternative that could also transform 3D interest points, the testing of 2D keypoints on 3D features in a virtual space was possible using 2D or 3D distance of points in the scene. The results of the experiment demonstrated that 2D repeatability could be emulated with 3D features. When conventional detectors were tested, the high degree of accuracy due to the virtual scene's controlled conditions that eliminated all noise, highlighted aspects of window processing that had not normally been noticable, and also reliably reflected performance behaviours observed by SURF, the introduction of noise to the scene was also reliably detected as a measurable degradation to the performance of the detectors being tested. This demonstrated the reliable repeatability evaluation of 2D interest points with a simulated 2D environment, and 3D models without requiring the detector to have any knowledge of the 3D scene itself.

7.2.2 Informedness

Chapter 4 investigated the utilisation of informedness as a new form of performance evaluation. The goal of utilising the characteristics of the virtual space to perform better analysis was achieved by utilising informedness to measure the repeatability performance of 2D keypoints when the distances of any two closest points were measured without and with the depth of the scene before 2D before ϵ thresholds were applied. This in turn enabled the performance of 2D detections (the classifier) to be compared to 3D detection (the ground truth) at each ϵ.

To investigate the performance of 2D and 3D detectors with Schmid repeatability, 9 conventional detectors were tested with a single 3D model and with 12 3D models (12models). The results of these tests were compared in terms of performance at each ϵ threshold. The results of the experiment indicated that 2D-based repeatability had better Schmid repeatability at each ϵ than 3D in most instances. Given that 3D ground truth was well established, this raised the question of whether occlusion could be distorting performance. Transforms were deliberately chosen to minimise performance, and examination of the results for each individual transform proved that 2D performance was higher in instances of transforms that could not cause occlusion. The experiment demonstrated that Schmid was misclassifying points when not utilising depth to find the closest repeated point pair.

To investigate the difference between Schmid and informedness, 9 conventional detectors were tested with a single 3D model and a 12model dataset. The results were compared in terms of performance at each ϵ threshold. The results of the experiment indicated that informedness was able to discriminate between 3D and 2D performance, and that, unlike Schmid, it could determine the most optimal ϵ threshold for each detector. In many cases, the optimal threshold was higher than the Moore neighborhood of pixels of $\epsilon = 1.5$ in some cases. Informedness also provided a better comparison between 2D and 3D by comparing 2D performance relative to 3D.

To investigate the informedness performance difference between RGB and greyscale with a 2D image, trained classifiers (see Section 7.2.3) were optimised with the use of RGB channels, or greyscale image data. The results of the experiment, named analysis A, demonstrated that classifiers trained using RGB had a statistically significant performance advantage compared to greyscale. This was apparent in both Schmid and informedness results, and demonstrated that informedness can be applied to other aspects of the virtual space beyond 2D/3D training comparisons. In addition, training using color was tested with a dataset that used color texturing in only half of the models used, showing that it generalised well. Informedness also highlighted that $\epsilon = 2.0$ had the best performance trade-off.

7.2.3 GP Classifier Optimisation via Virtual Ground Truth

The goal of testing the viability of classifier optimisation via GP was successfully achieved in Chapter 5 through the implementation of GP in conjunction with the STEIPR with managed the virtual space. STEIPR could translate a classifier's syntax tree provided by the GP algorithm, and process detected 2D keypoints via the virtual space in a manner similar to conventional detectors already tested.

To investigate the optimisation performance of classifiers in a virtual space, a 2D image was utilised for training in both 2D and 3D. The experiment results, relevant to analysis B, demonstrate that performance was identical and suggested that the use of depth for 3D operated as expected, as well as showing that the threshold $\epsilon = 2.0$ had the best performance trade-off.

To investigate the optimisation of training RGB classifiers with a larger dataset, the 12model dataset was used. The experiment results, relevant to analysis C, demonstrate that testing informedness with a singular 2D image produced unpredictable results, due to insufficient data.

To investigate the optimisation of training 2D/3D classifiers with a larger dataset, the 12model dataset was used. The experiment results, relevant to analysis D, show that testing informedness with a singular 3D model did not produce unpredictable results as analysis C did; this approach did, however, raise the possibility of a lack of generalisation. The informedness results demonstrated better performance than that of conventional detectors, but were arguably inconclusive due to the fact that it was not possible to restrict the number of max points per scene for conventional detectors in the same way that this can be done for evolved classifiers.

To investigate the optimisation of training 2D/3D classifiers with better generalisation, a single 3D model and the larger 12model dataset for testing. The experiment results, relevant to analysis E, demonstrated that performance was similar to the performance observed in analysis D and training time was much faster. The results also highlighted that $\epsilon = 2.0$ still had the best performance trade-off.

To investigate the optimisation of training 2D/3D classifiers with a number of maximum points per scene which was more similar to that for conventional detectors, the max points per scene was raised from 500 to 2000. The experiment results, relevant to analysis F, showed that repeatability performance compared to analysis E was considerably better. Both Schmid and informedness reflect this, but for informedness, the best trade-off was only from $\epsilon = 0.5$ to $\epsilon = 2.0$.

To investigate the optimisation of training 2D/3D classifiers with shadows, the light was positioned at the top left and top right. The experiment results, relevant to analysis G and H, showed that the non-symmetrical 3D models affected performance such that both Schmid and informedness were affected differently, with Schmid's interquartile range being skewed up/down, and the interquartile informedness range being tighter/wider.

To investigate the optimisation of training 2D/3D classifiers with higher resolution, the image resolution was increased from 300px to 600px. The experiment results, relevant to analysis I, demonstrated that performance in 2D was statistically different to performance in 3D, and indicated that the use of GP optimisation with depth data could affect how classifiers are optimised in an emergent manner. When compared to conventional detectors, the best evolved classifiers outperformed all but Harris at $\epsilon = 1.5$ and $\epsilon = 2.0$, but did so with fewer points compared to Harris, which was unrestricted in terms of the maximum number of points per scene.

A resolution of 1000px was also investigated to test the optimisation of training 2D/3D classifiers. The experiment results, relevant to analysis J, showed poorer results compared to earlier tests, and conventional detectors under similar testing conditions. The relative weakness of these results was likely due to the limitations of the max points per scene, in conjunction with the dataset having degraded detail at the much higher resolution.

7.3 Limitations

What follows is an outline of limitations experienced in undertaking the implementation and analysis of this dissertation's research.

7.3.1 Occlusion

For the assessment of IP performance in 3D scenes, a number of approaches were utilised to eradicate, or at least minimise, occlusion. To more firmly address occlusion, however would require access to the depth buffer at the shader level, and would be non-trivial to implement. As such, this limited the degree of complexity of scene transformations as well as the complexity of the scene itself, as tight control of the state of the scene was required to correctly utilise repeatability in this research.

7.3.2 Computational Cost

Computational cost played a factor in undertaking testing, which limited the training of classifiers, such that modest image sizes increased training time considerably, even after classifier processing was entirely implemented in C++, as shown in (Table 5.5 and 6.4). Even under ideal testing conditions, and fully utilising multiple machines in a parallel manner to spread the workload, the training length for each test was considerable. To lessen the training length, I used fewer training models and more testing models, as early tests did not drastically affect performance and testing was considerably faster.

7.3.3 Interest Point Constraint

Another factor was the maximum number of points per scene (max points) which also increased processing time due to the measuring of point distances, which was computationally expensive, though not as much. Increasing the number of points, as seen in experiment F (6.3.2), did not offer any significant benefit from an analytical or performance perspective, but probably shouldn't be ruled out if image sizes were increased. This was not pursued due to limited time available and would not necessarily have helped in post training analysis. The lack of conventional detectors that were available with limitations on the points returned (as well as ordering by the strength of their IP response) was also mixed at best, and non-existent in most cases, so they did not lend themselves as appropriate candidates for analysis based on what was being tested, and the statistical rigor required.

As such, I was limited by existing implementations, as well as the internal design of the algorithms themselves that may create biased, or misleading results due to the manner in which the scene is processed internally, eg. points returned from thrid party implementations could have been ordered top left, to bottom right, so any restriction on points could result in misleading analysis as it would “cut off” part of the scene if the max points restriction were applied.

7.3.4 Interest Point Algorithm Diversity

In order to have a sufficiently adequate number of candidate detectors to statistically analyse, the limited pool of avaiable conventional detectors meant that most of our testing had to be done with GP-based variants. This was especially apparent in cases where I wished to test other aspects of the virtual scene using color channels. The lack of appropriate 3rd party detectors that conformed to the rigorous statistical and functional demands meant, that in most cases we were limited to rapid prototyping via GP as inhouse implementation of existing algorithms would have been intensely time consuming, difficult, or in many cases not possible at all.

7.4 Final Summary

In summary, the the 3D modelling paradigm has brought about useful techniques that can facilitate the evaluation of the performance and tracking of 2D keypoints from conventional detectors, with Harris being almost always very competitive at $\epsilon = 1.5$ and $\epsilon = 2.0$. While Harris and Fast sometimes find lots of trackable points that may be useful in certain contexts illustrated by the heatmaps in appendix C, there is clearly a bias towards certain texturing, rather than a generalisation across all models that would find consistently “interesting” points. The informedness measure has also proven to be an important extension of the approach to evaluate point repeatability. As the name “informedness” implies, the evaluation metric often provides sharper, more informed and more meaningful information than the Schmid repeatability measure, which does not incorporate as much information about the scene due to insufficienly reliable and accurate ground truth. It is also shown with 95% confidence, that under certain testing conditions (see 6.3.5), Informedness and Schmid's performance diverges due to Schmid's lack of incorporation of type II errors that can only be verified with a ground truth. An advantage of the GP approach is that it controls the number of interest points better than many of the standard detectors, can test various forms of interest point optimisation unavailable in existing detectors, such as colour, and controls against potential bias of these detectors, which often detect far too few or far too many points. The utilisation of GP for assistance in analysing informedness in particular, has been intrinsic to establishing a consistent method of isolating all independent variables, and controlling for numbers of points per scene. Use of GP has also assisted in empirically demonstrating that GP training can optimise for

3D environments (and potentially real world scenes) with the assistance of depth of virtualised 3D scenes passively, and without deliberate intervention.

7.5 Future Work

This section details some recommended areas deserving of future focus beyond the research in this thesis.

1. "Back pane culling" for interest points.

   Due to the fact that it was difficult to identify occluded points, one approach that could be used mirrors the approach called "back pane culling" in 3D graphics. This approach utilises an algorithm to find the closest vertex when rendering a scene and ignore all other vertexes in that pixel in order to reduce rendering time. The use of the depth buffer and vertex depth is native to shaders, but unfortunately not accessible outside of the graphics card, due to the complexity of the models used. Being able to precisely identify the location of a vertex in Euclidean space based on the pixel position where an IP appears is necessary if occlusion is to be 100% avoided. Developing a tool that could utilise vertex depth at the shader level in conjunction with the pixel position and depth would help identify points that are occluded by the model in the scene.

2. More tests at 600px and other configuration changes such as 2k max points and RGB terminals.

   The research in this thesis, supports the completion of further tests that can be undertaken to better explore the conditions under which 3D training will outperform 2D training. Increasing of the max points per scene, as well as running other tests at 600px, would help build a stronger argument that virtual optimisation of classifiers is not properly leveraging the potential advantages of the utilistation of scene depth by being restricted to 2D data alone when training.

3. Augmenting the fitness function with informedness.

   Based on the very promising informedness results, and the fact that this metric uses the same 0 to 1 range as Schmid, it would be worthwhile to utilise informedness from within the GP training system rather than as a post-analysis tool. Informedness also has additional benefits where classifiers can be optimised based on the best trade-off ϵ, rather than at the default $\epsilon = 1.5$, which the results of this thesis have empirically demonstrated is questionable.

4. Larger training datasets.

   Though the use of larger training datasets results in considerably longer training times, there is still an argument to be made that larger datasets will help limit overtraining and optimise for more robust points.

5. Re-implementation of conventional detectors that permits limiting the number of strength-ordered points.

   To get a genuine comparison between evolved classifiers and conventional detectors, an emphasis on eliminating independent variables such as numbers of points per scene, as well as ordering by point strength is desirable. Under the current conditions, this was not feasible, due to the detectors being third party implementations that were inflexible and largely unmodifiable. The necessity of working in native C++ also limited the availability of alternatives.

6. Improved virtual scenes.

   More complex scenes are also desirable if classifiers are to be optimised for real-world applications.

Appendix A

Dataset Samples

This section shows samples of the datasets used in the testing, and training of interest point operators.

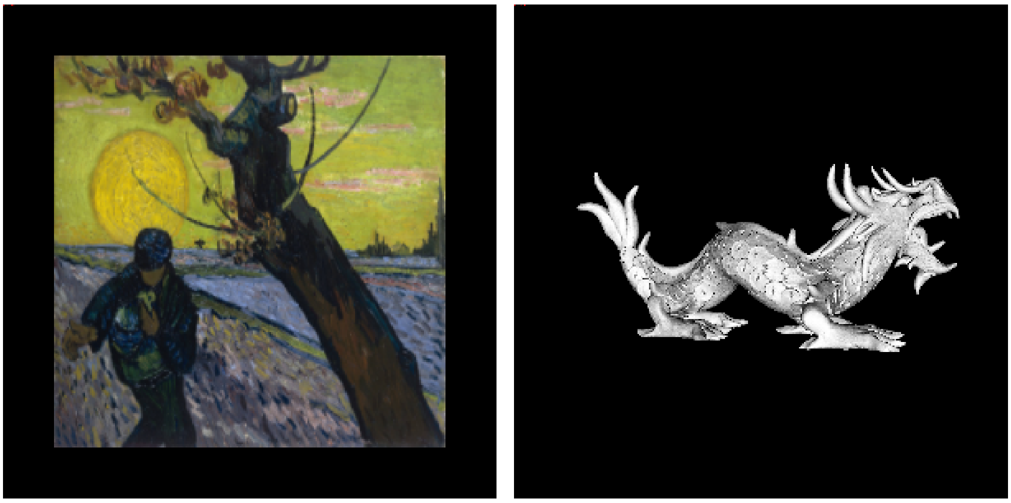
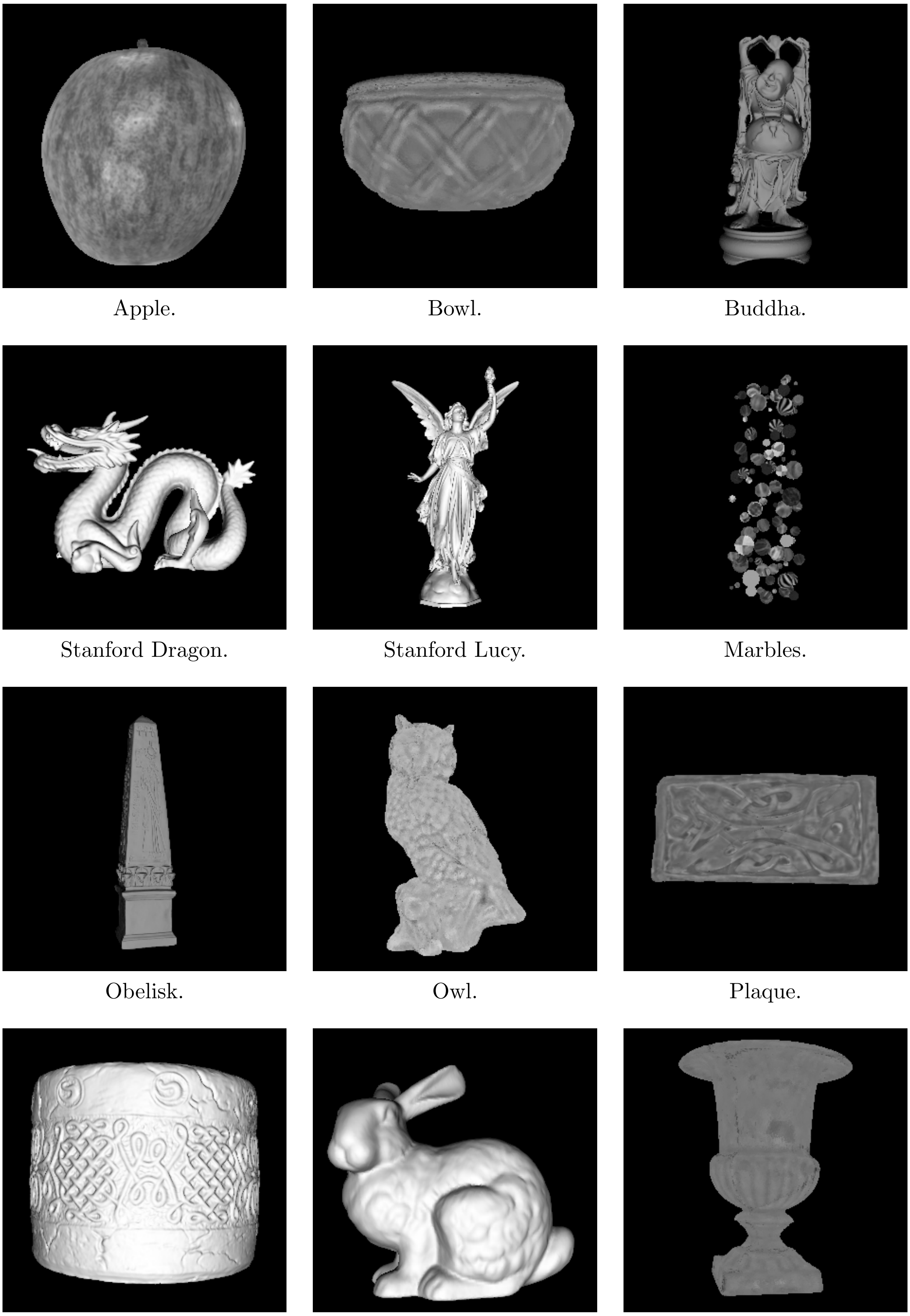


Figure A.1: Reference transform of Sower Image and asian dragon.



Apple.

Bowl.

Buddha.

Stanford Dragon.

Stanford Lucy.

Marbles.

Obelisk.

Owl.

Plaque.

Pot.

Stanford Bunny.

Vase.

Figure A.2: Reference transforms of 12 model dataset.

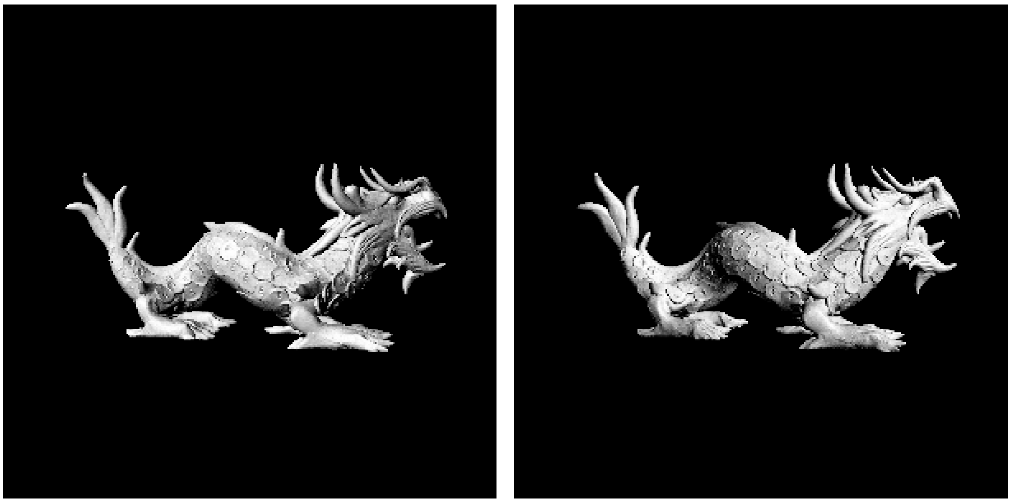
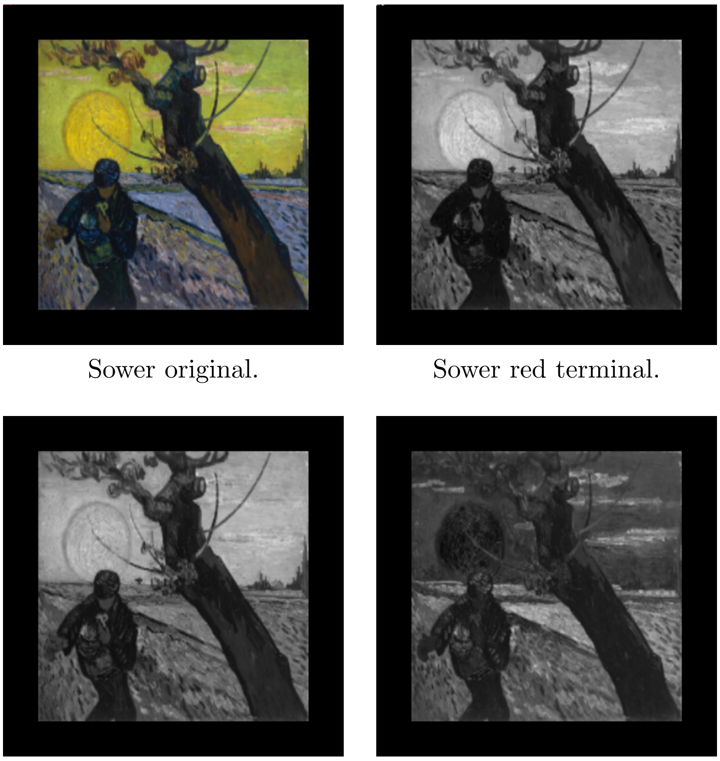


Figure A.3: Reference transform of light positions top left (left), and top right (right) for asian dragon model.



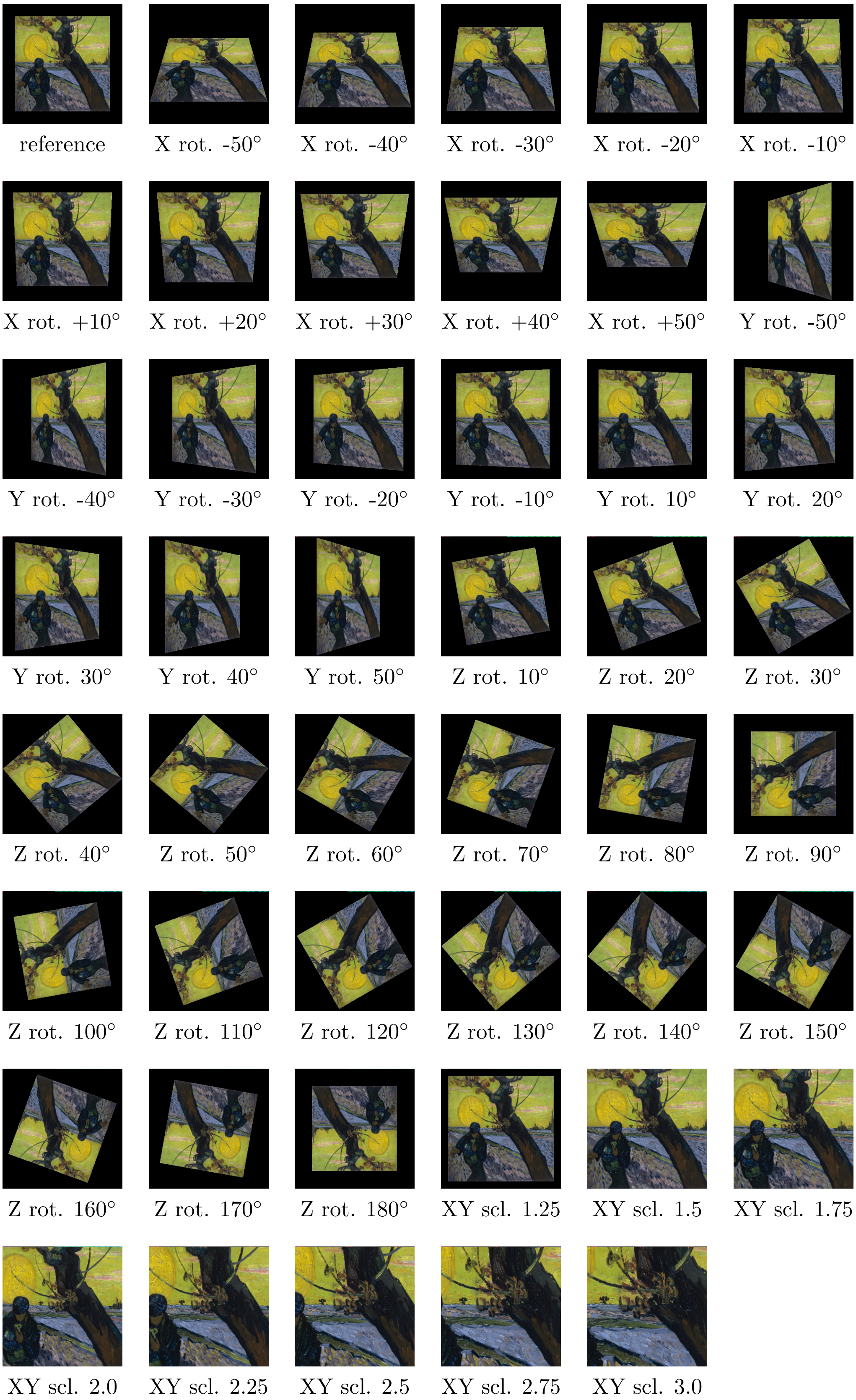
Sower original.

Sower red terminal.

Sower green terminal.

Sower blue terminal.

Figure A.4: Sower's Red, green, and blue terminal samples as intensity images.



reference, X rot. -50°, X rot. -40°, X rot. -30°, X rot. -20°, X rot. -10°

X rot. +10°, X rot. +20°, X rot. +30°, X rot. +40°, X rot. +50°, Y rot. -50°

Y rot. -40°, Y rot. -30°, Y rot. -20°, Y rot. -10°, Y rot. 10°, Y rot. 20°

Y rot. 30°, Y rot. 40°, Y rot. 50°, Z rot. 10°, Z rot. 20°, Z rot. 30°

Z rot. 40°, Z rot. 50°, Z rot. 60°, Z rot. 70°, Z rot. 80°, Z rot. 90°

Z rot. 100°, Z rot. 110°, Z rot. 120°, Z rot. 130°, Z rot. 140°, Z rot. 150°

Z rot. 160°, Z rot. 170°, Z rot. 180°, XY scl. 1.25, XY scl. 1.5, XY scl. 1.75

XY scl. 2.0, XY scl. 2.25, XY scl. 2.5, XY scl. 2.75, XY scl. 3.0

Figure A.5: Transforms used during evolutionary testing, consisting of X,Y and Z axis rotation and XY scaling. In this example the sower image is used.

Appendix B

Conventional Detector Tests: Analysis D-J

This appendix contains tests of conventional detectors that use the same testing configurations used in Chapter 5 and 6. Analysis E and F utilise the same figures which is due to these analyses' only difference being the maximum points per scene. The remaining Figures in this Appendix should be utilised for informational purposes and not direct analysis however as the lack of a max point limit could bias results, the only exception being the Figures for E and F which aim to lessen this potential bias. As noted in Section 6.3.2, increasing the max point total per scene did in fact increase performance all be it, not the maximums, but this demonstrated that if the conventional detectors were similarly constrained, their performance could be reduced also. In figures B.1 and B.2 the conventional detectors used have no strict limit (due to the inflexibility of the detectors themselves and lack of alternative implementations), and ranges from 0 to 2000 points per scene, whereas analysis E and F is restricted to a maximum of 500 and 2000 respectively. Due to the lack of utilisation of color, the tests for A and C are not applicable, and B has been omitted due to it not being applicable for further analysis.

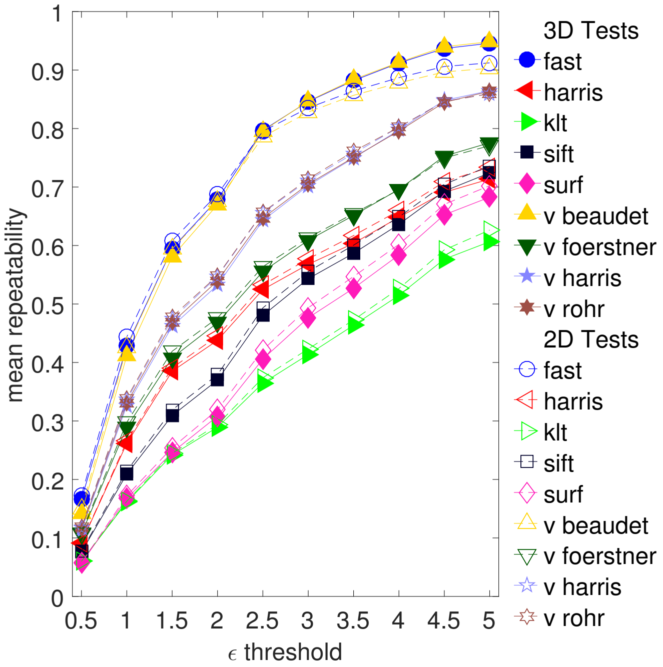


Figure B.1: Tested using Analysis D Configuration: 2D (open dashed) overlayed versus 3D (solid) Schmid repeatability of conventional detectors using the asian dragon model.

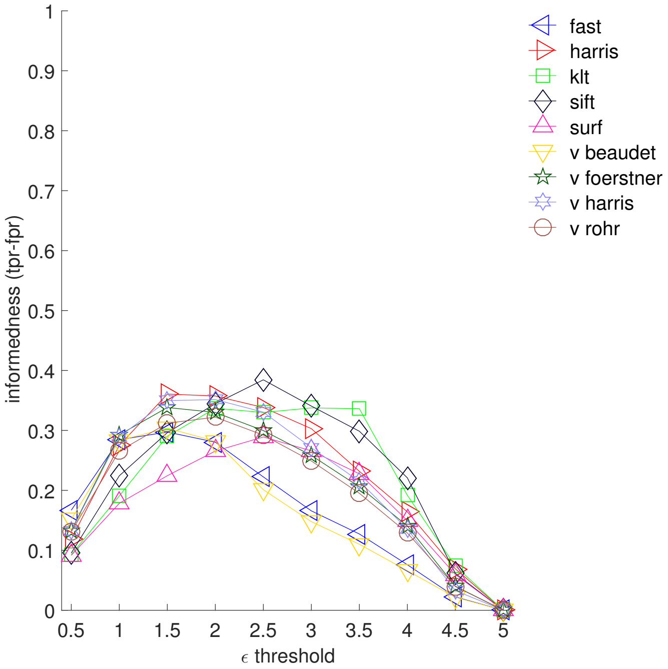


Figure B.2: Tested using Analysis D Configuration: 2D/3D Informedness of conventional detectors using the asian dragon model.

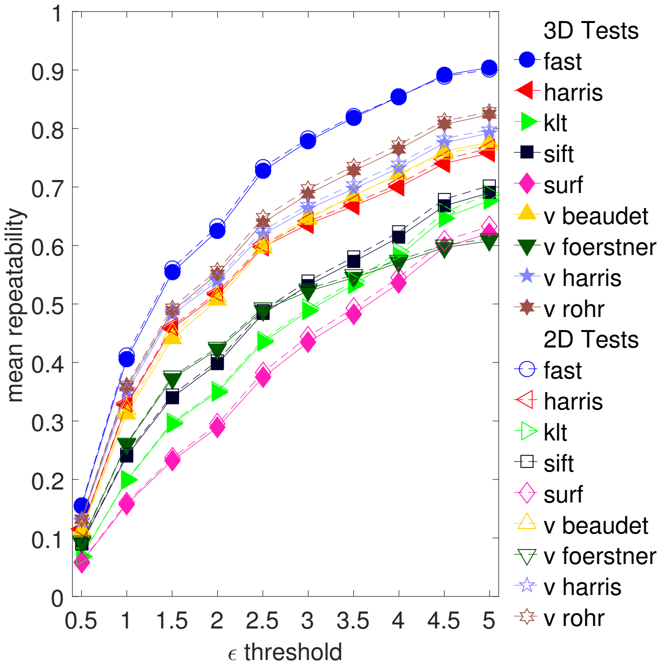


Figure B.3: Tested using Analysis E/F Configuration: 2D (open dashed) overlayed versus 3D (solid) Schmid repeatability of conventional detectors using the 12model dataset.

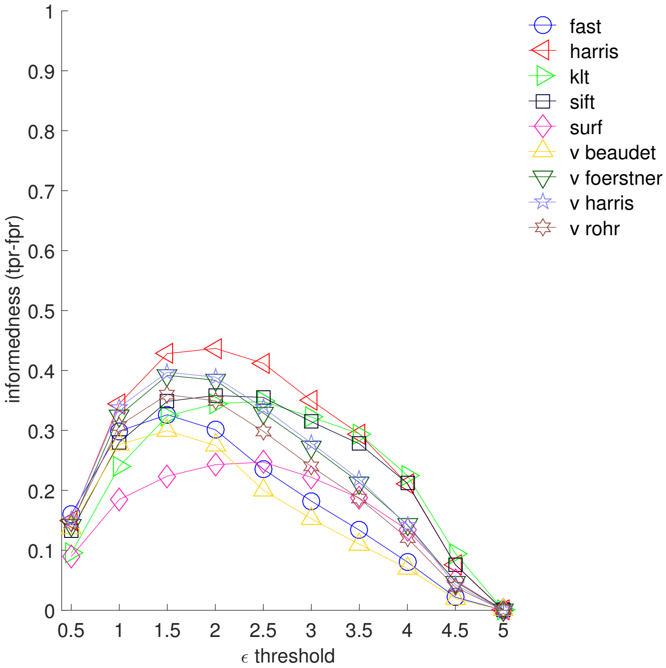


Figure B.4: Tested using Analysis E/F Configuration: 2D/3D Informedness of conventional detectors using the 12model dataset.

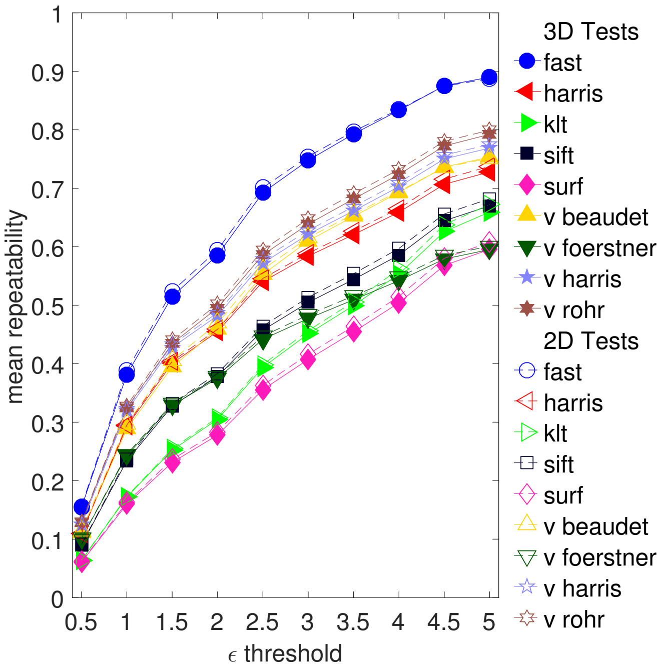


Figure B.5: Tested using Analysis G Configuration (light top left): 2D (open dashed) overlayed versus 3D (solid) Schmid repeatability of conventional detectors using the 12model dataset.

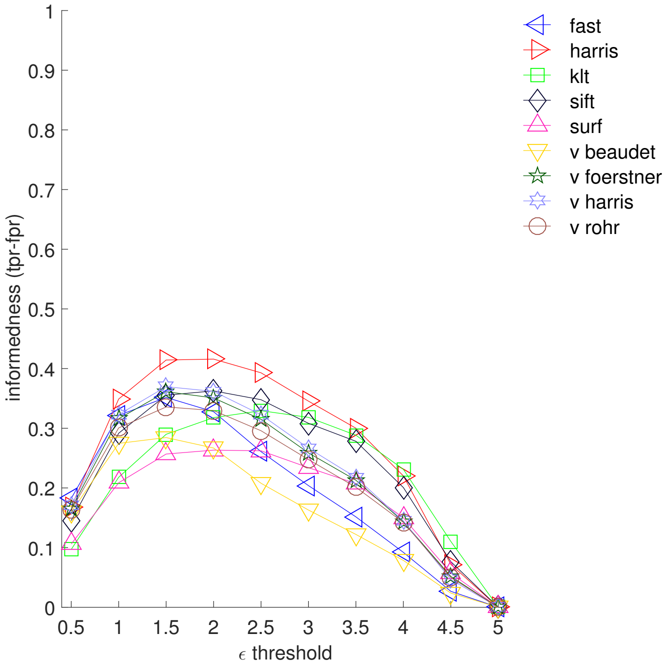


Figure B.6: Tested using Analysis G Configuration (light top left): 2D/3D Informedness of conventional detectors using the 12model dataset.

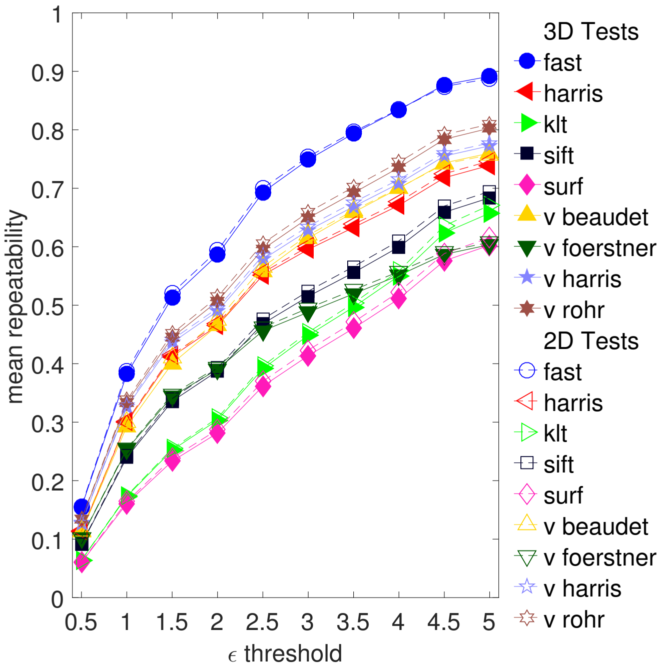


Figure B.7: Tested using Analysis H Configuration (light top right): 2D (open dashed) overlayed versus 3D (solid) Schmid repeatability of conventional detectors using the 12model dataset.

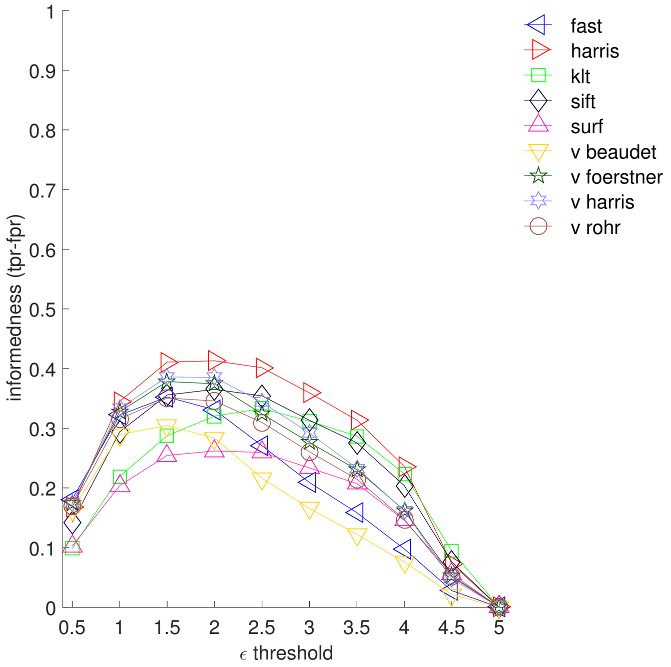


Figure B.8: Tested using Analysis H Configuration (light top right): 2D/3D Informedness of conventional detectors using the 12model dataset.

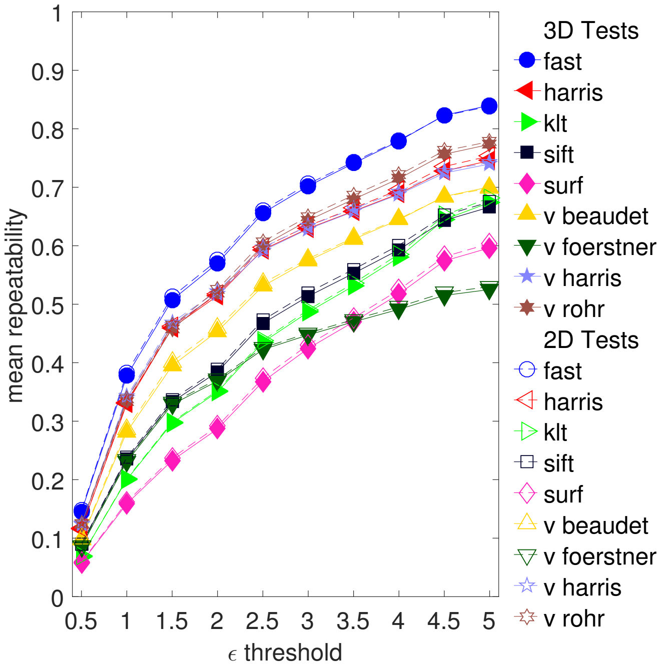


Figure B.9: Tested using Analysis I Configuration (600px image): 2D (open dashed) overlayed versus 3D (solid) Schmid repeatability of conventional detectors using the 12model dataset.

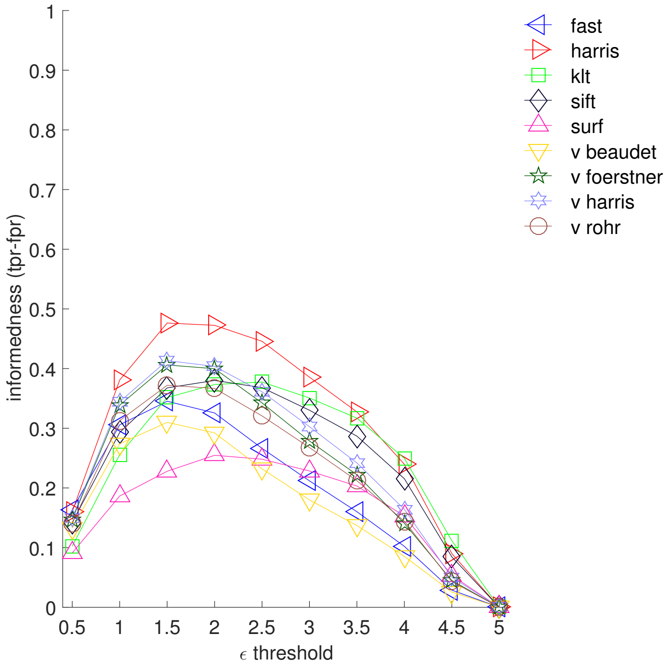


Figure B.10: Tested using Analysis I Configuration (600px image): 2D/3D Informedness of conventional detectors using the 12model dataset.

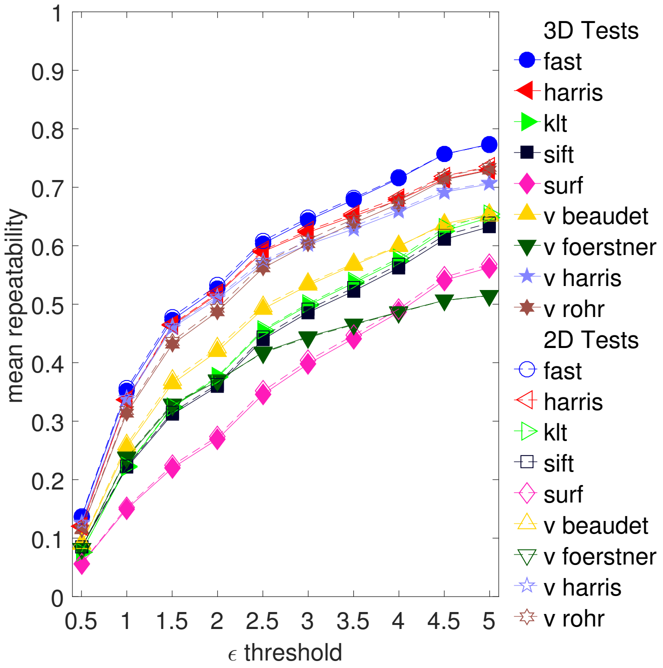


Figure B.11: Tested using Analysis J Configuration (1000px image): 2D (open dashed) overlayed versus 3D (solid) Schmid repeatability of conventional detectors using the 12model dataset.

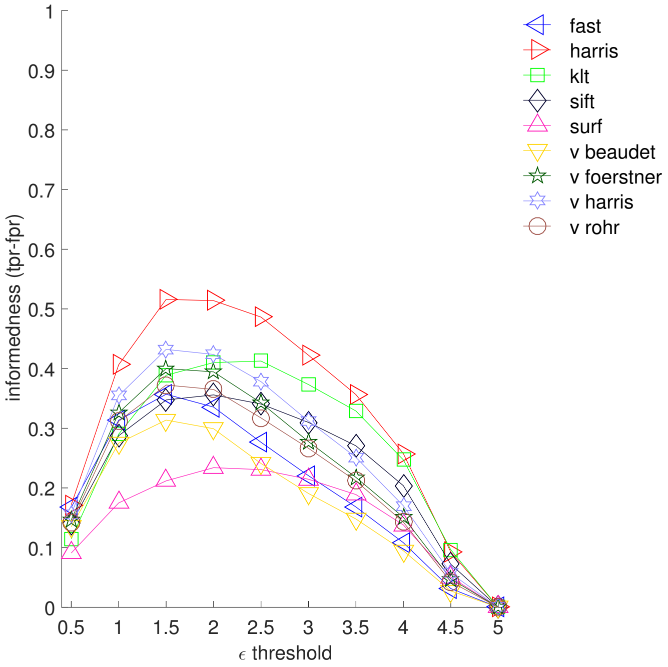


Figure B.12: Tested using Analysis J Configuration (1000px image): 2D/3D Informedness of conventional detectors using the 12model dataset.

Appendix C

Informedness: True Positive and False Positive Heatmaps for Conventional Detectors

The following heatmaps are representations of repeated 2D/3D points that have been classified as the same (True Positives) in 2D and 3D, or Existing in 2D, but not in 3D (False Positives). This represents the counts of these TP and FP instances when comparing interest points. This information forms the basis of the Informedness graphs used in this dissertation. Each figure forms a pair for the purposes of informedness, and is calculated according to each training run/conventional detector tested. The heatmaps can be read somewhat similarly to Chapter 4 Figure 4.1, except the plot is tilted 90 degrees and represents each "box" in the heat graph, and rather than plots, the data is read according to the increase in TP/FP counts along the ϵ range 0.5-5.0, represented within the box left to right along the x axis, and the model affine transform from bottom to top along the y axis. This gives an overall representation of performance according to each model, and which returned more or less TP or FP for each detector.

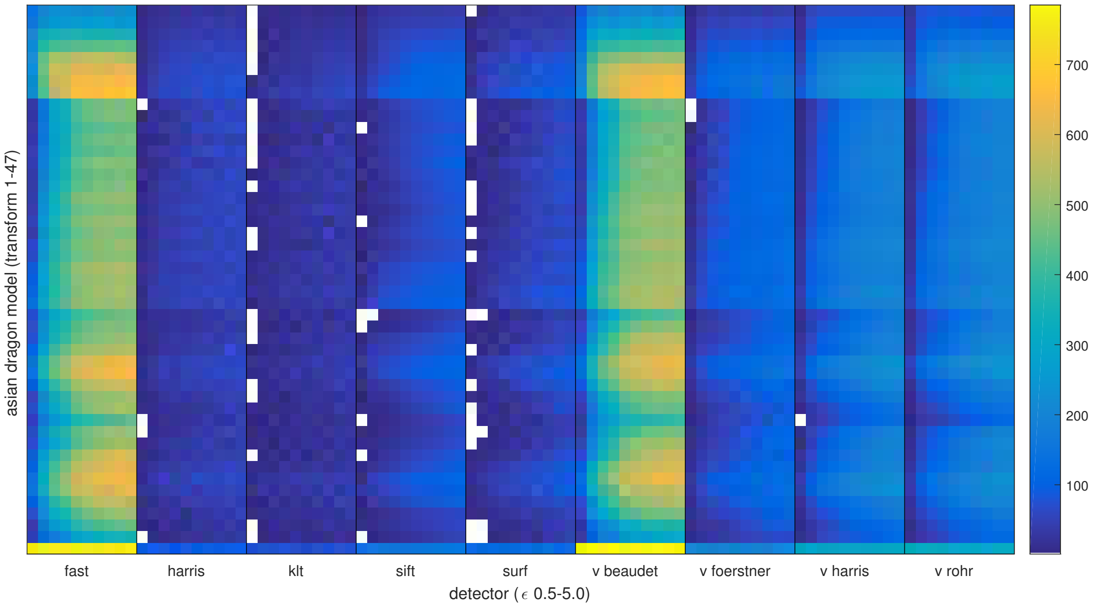


Figure C.1: Conventional Detectors (comparable to analysis D): Heatmap of TP repeated points.

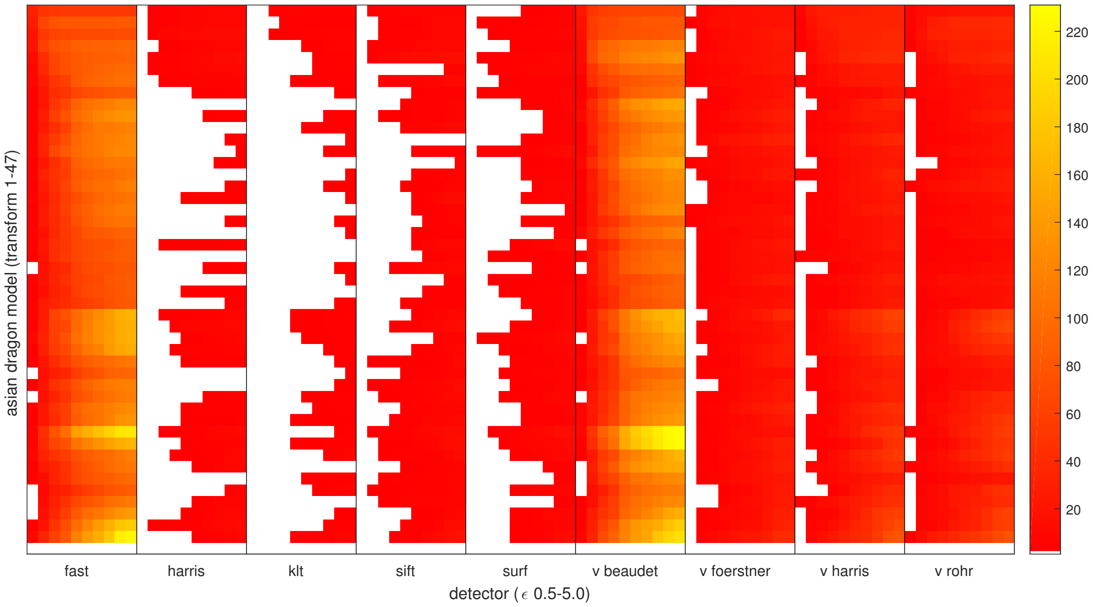


Figure C.2: Conventional Detectors (comparable to analysis D): Heatmap of FP repeated points.

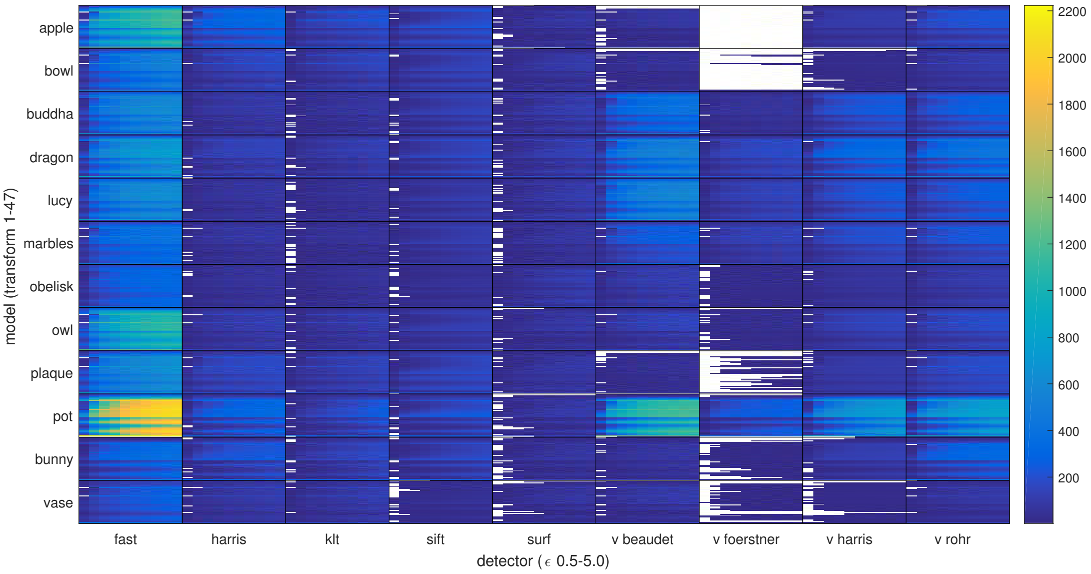


Figure C.3: Conventional Detectors (comparable to analysis E/F): Heatmap of TP repeated points.

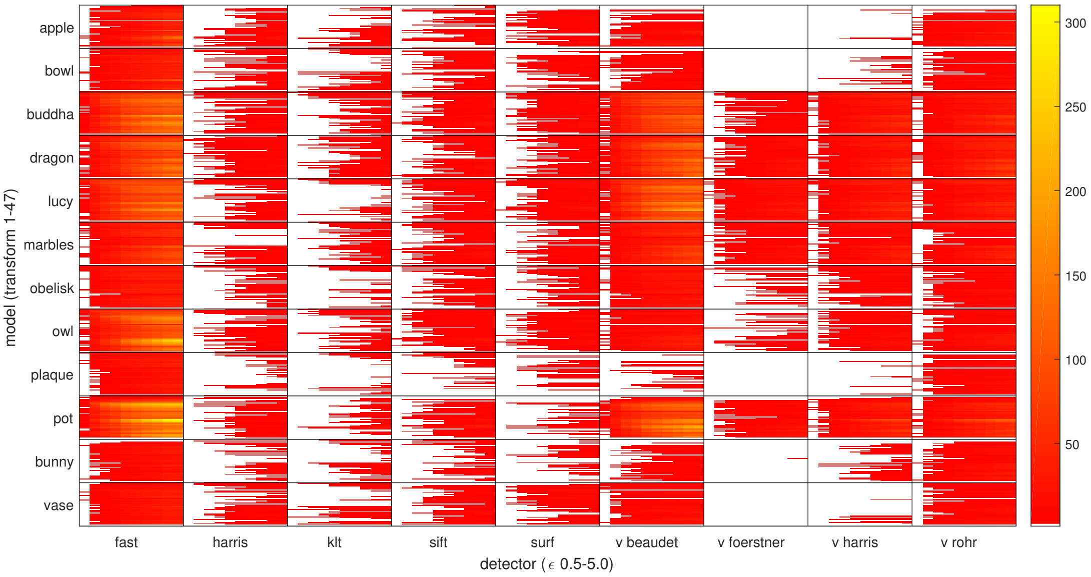


Figure C.4: Conventional Detectors (comparable to analysis E/F): Heatmap of FP repeated points.

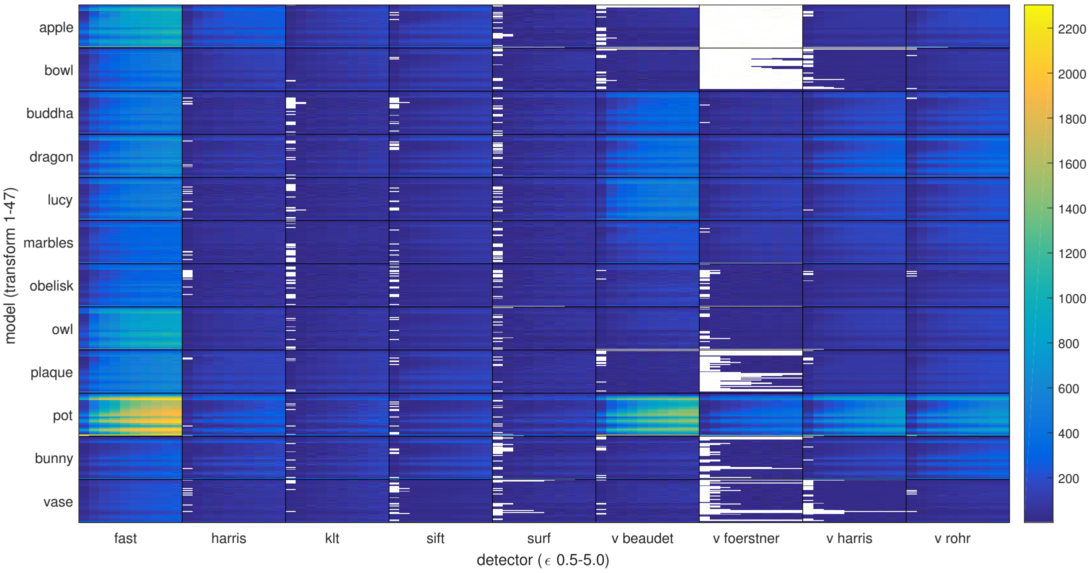


Figure C.5: Conventional Detectors (comparable to analysis G, light top left): Heatmap of TP repeated points.

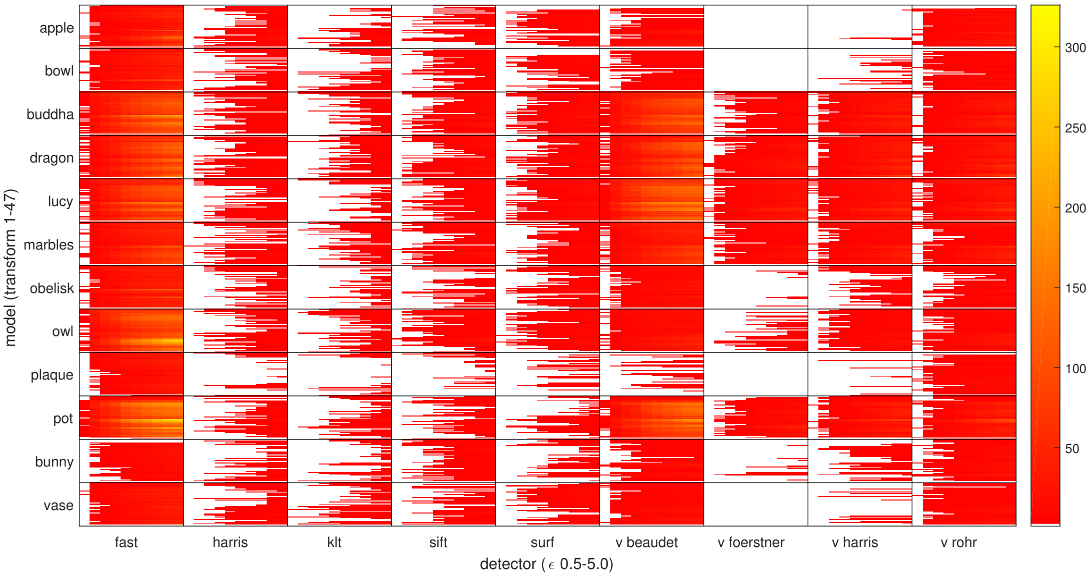


Figure C.6: Conventional Detectors (comparable to analysis G, light top left): Heatmap of FP repeated points.

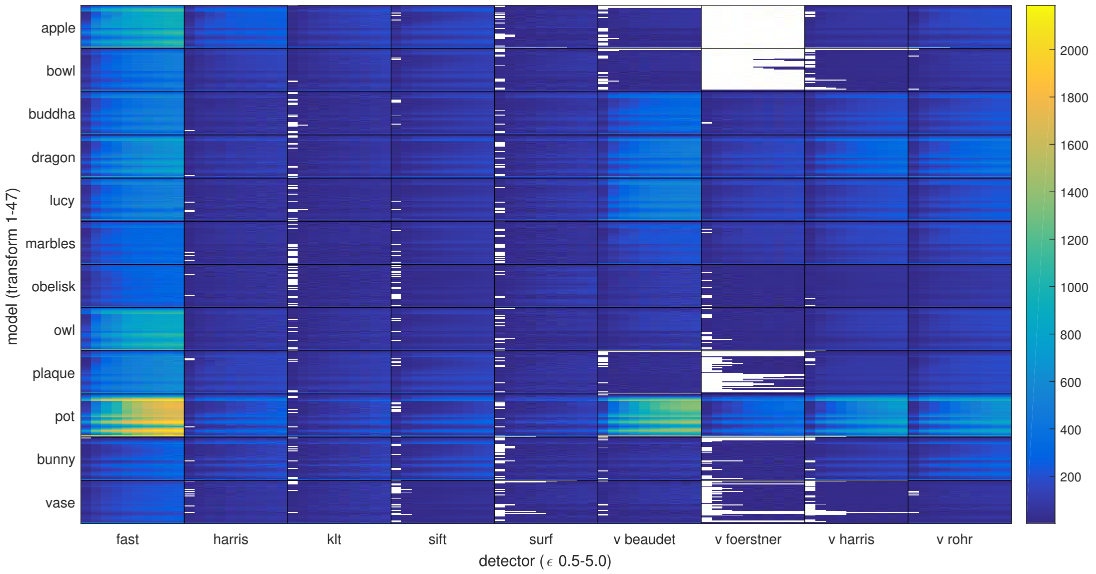


Figure C.7: Conventional Detectors (comparable to analysis H, light top right): Heatmap of TP repeated points.

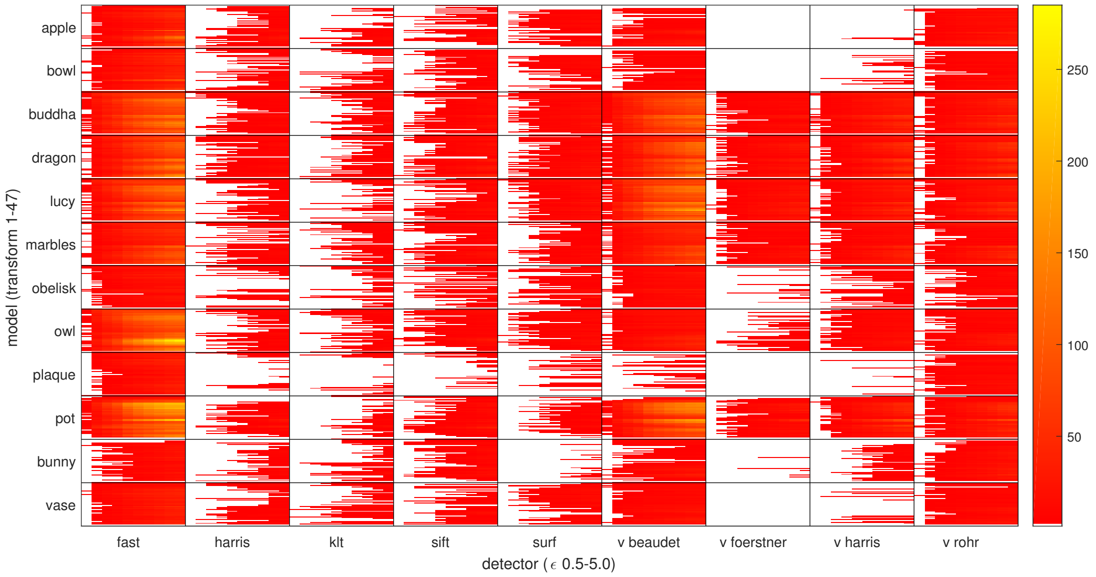


Figure C.8: Conventional Detectors (comparable to analysis H, light top right): Heatmap of FP repeated points.

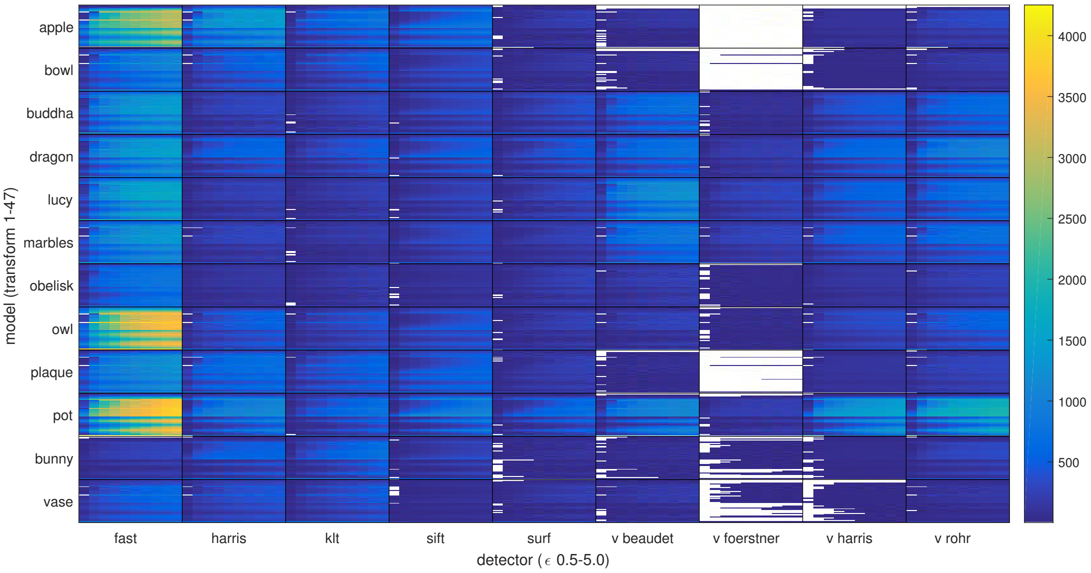


Figure C.9: Conventional Detectors (comparable to analysis I, 600px image): Heatmap of TP repeated points.

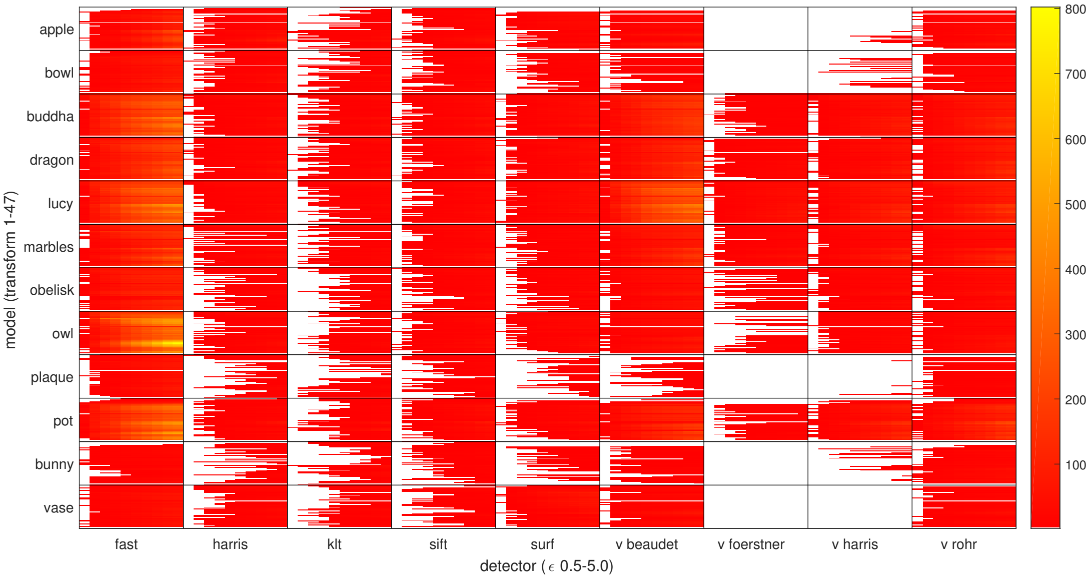


Figure C.10: Conventional Detectors (comparable to analysis I, 600px image): Heatmap of FP repeated points.

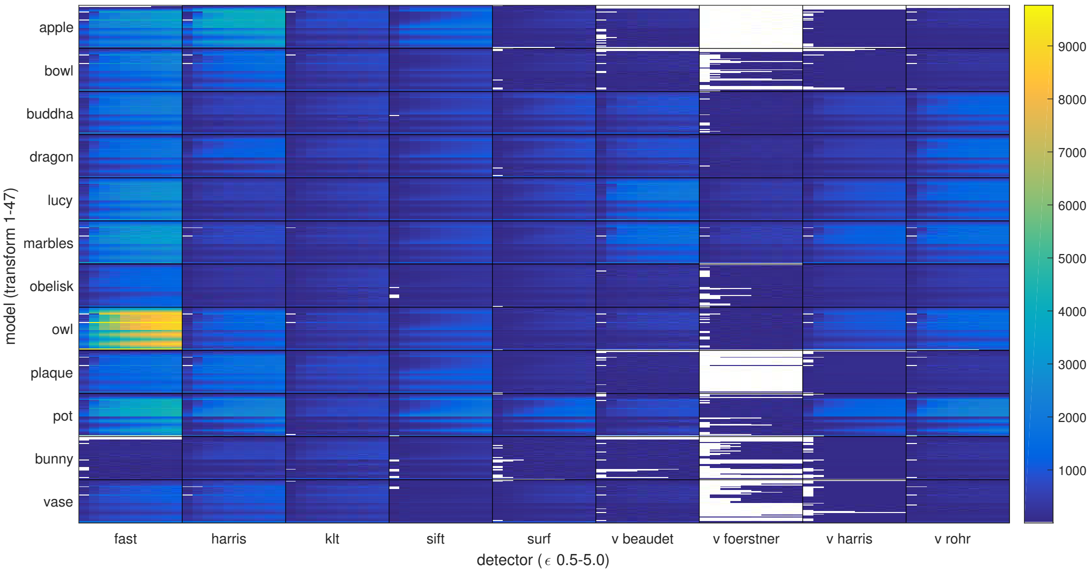


Figure C.11: Conventional Detectors (comparable to analysis J, 1000px image): Heatmap of TP repeated points.

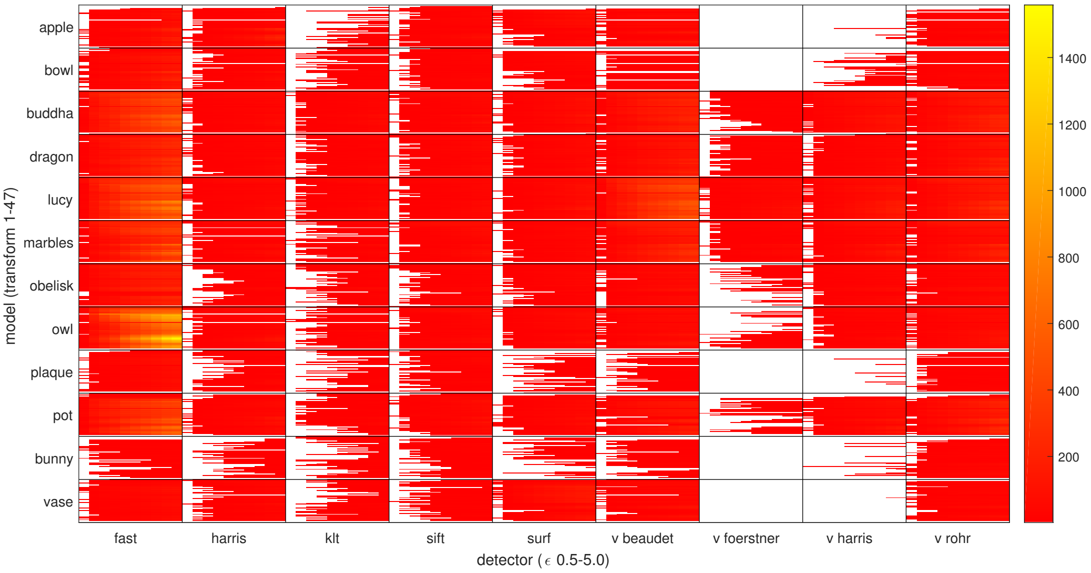


Figure C.12: Conventional Detectors (comparable to analysis J, 1000px image): Heatmap of FP repeated points.

Appendix D

Informedness: True Positive and False Positive Heatmaps For Evolved Classifiers

The following heatmaps represent the data used to determine Informedness and are ordered according to the analysis D-J.

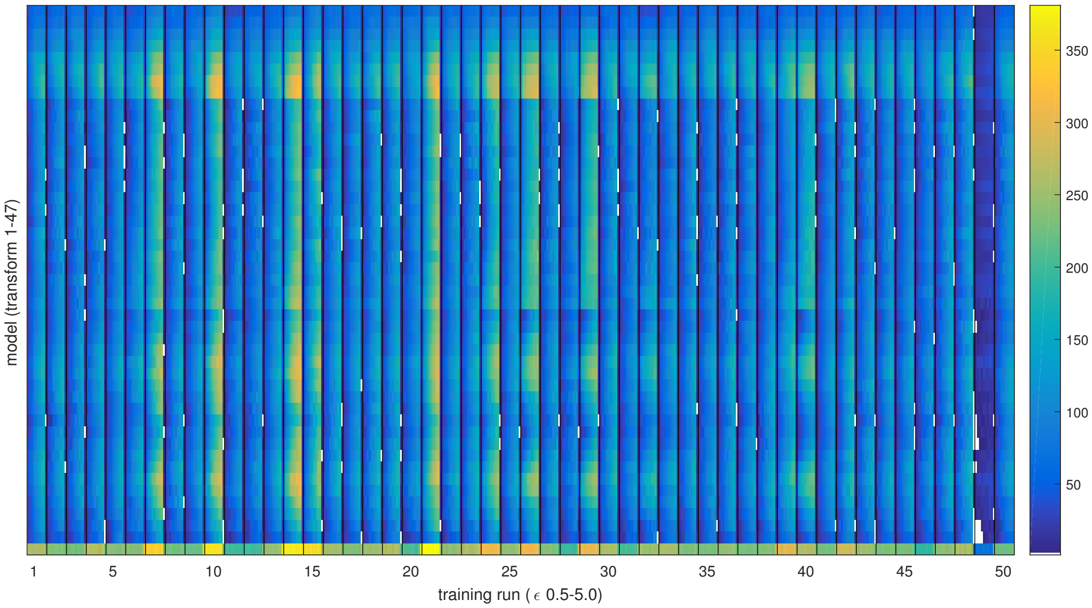


Figure D.1: Analysis D: Heatmap of TP repeated points for 3D-2D/RGB-GS training.

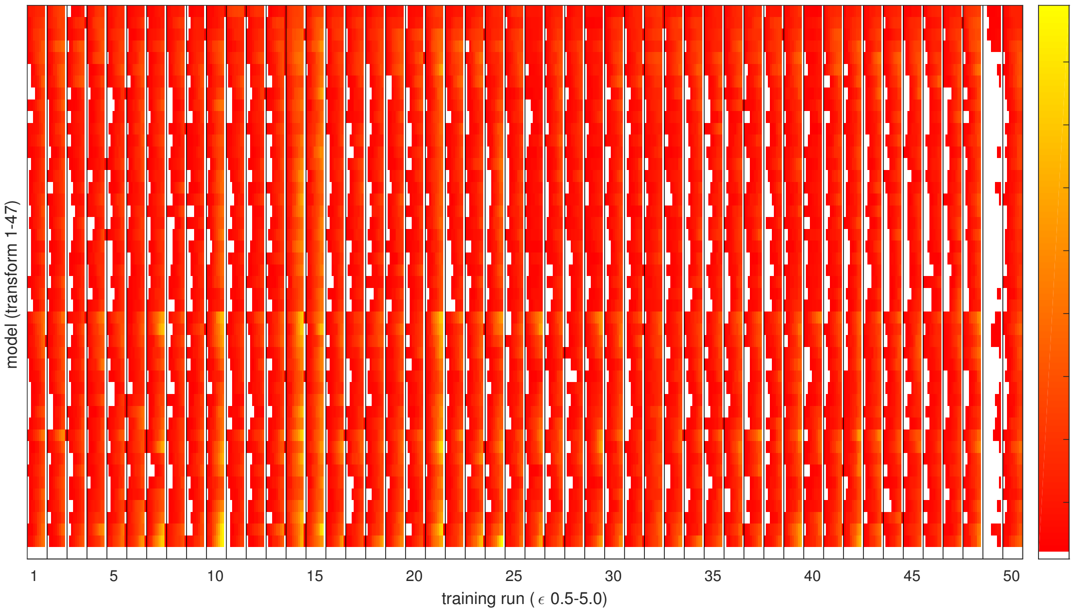


Figure D.2: Analysis D: Heatmap of FP repeated points for 3D training.

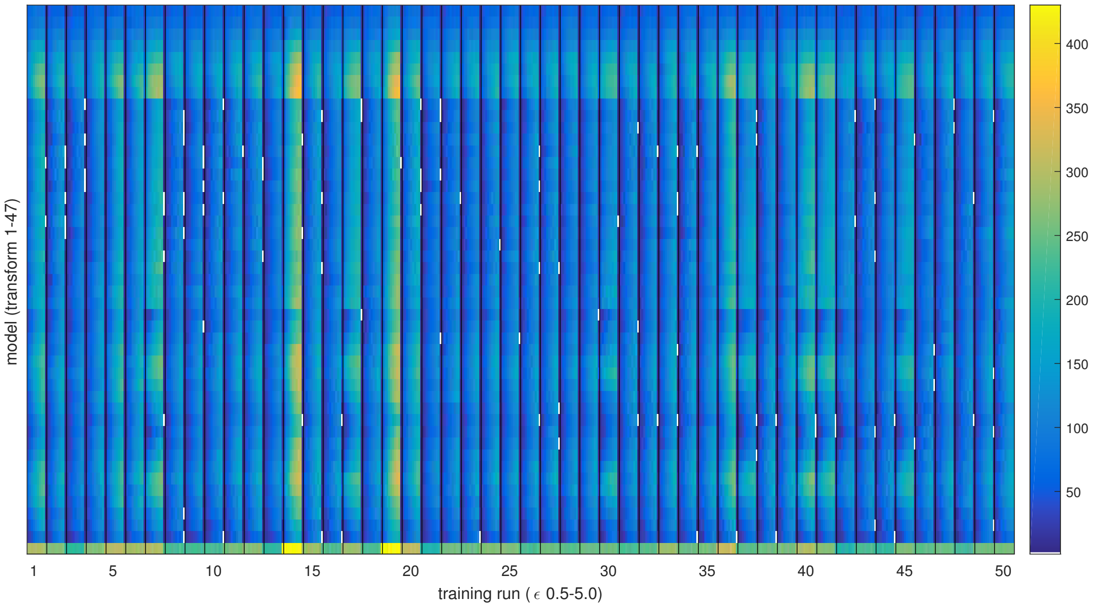


Figure D.3: Analysis D: Heatmap of TP repeated points for 2D training.

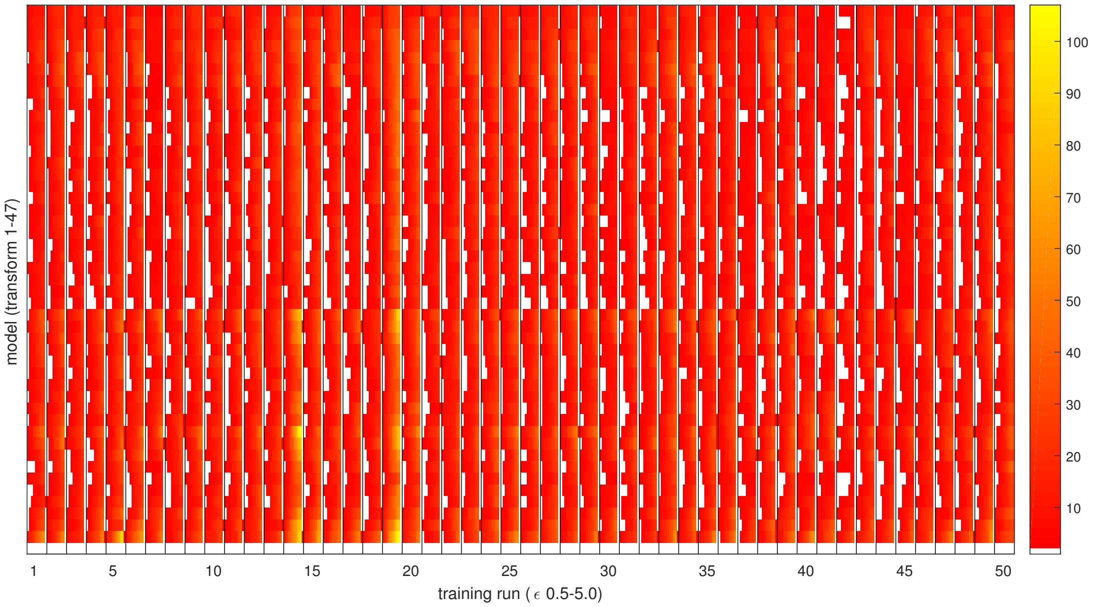


Figure D.4: Analysis D: Heatmap of FP repeated points for 2D training.

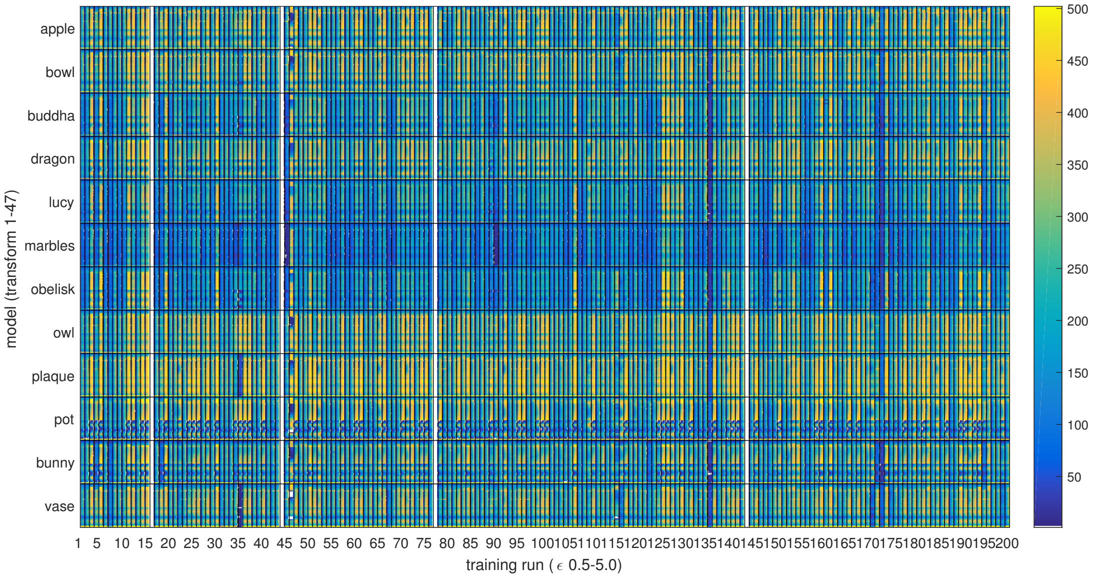


Figure D.5: Analysis E: Heatmap of TP repeated points for 3D training.

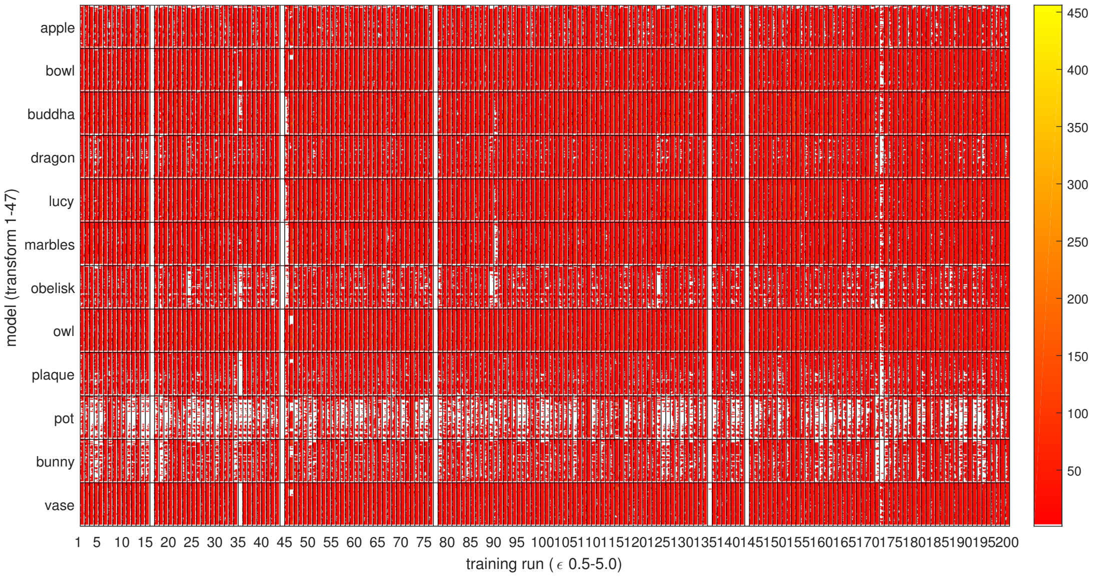


Figure D.6: Analysis E: Heatmap of FP repeated points for 3D training.

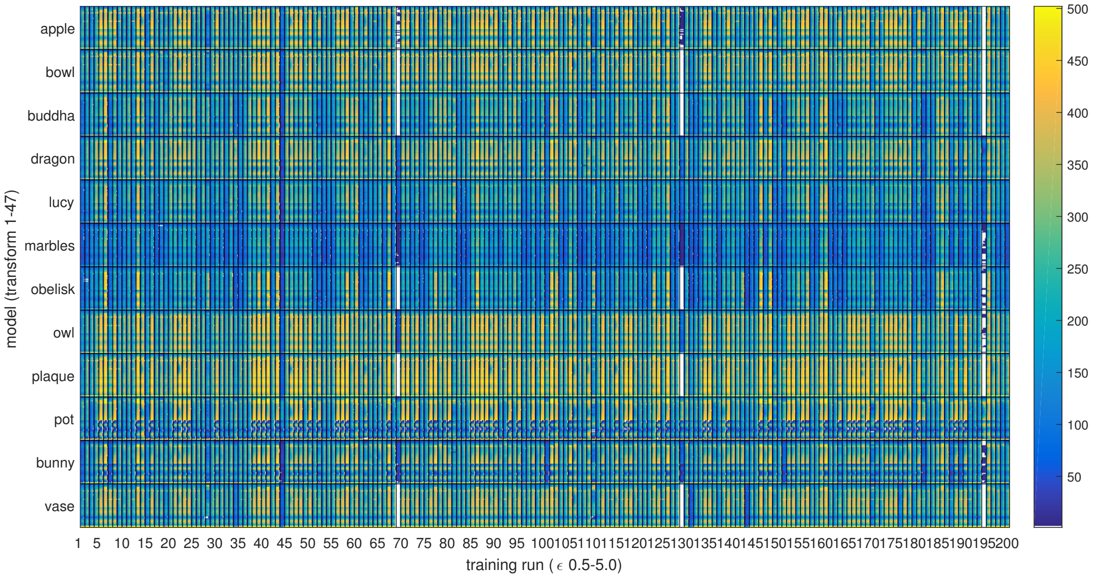


Figure D.7: Analysis E: Heatmap of TP repeated points for 2D training.

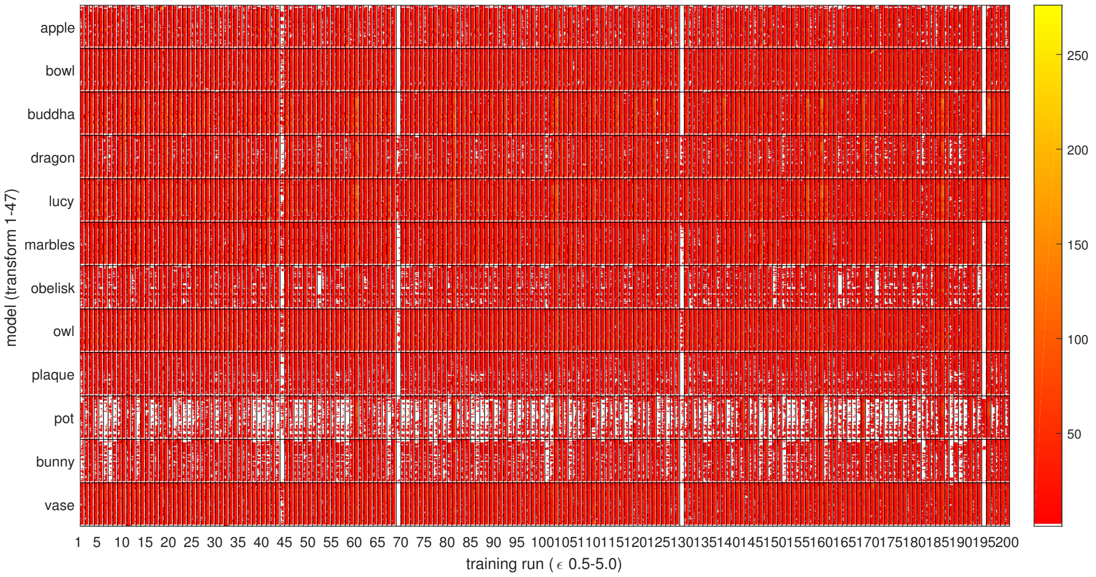


Figure D.8: Analysis E: Heatmap of FP repeated points for 2D training.

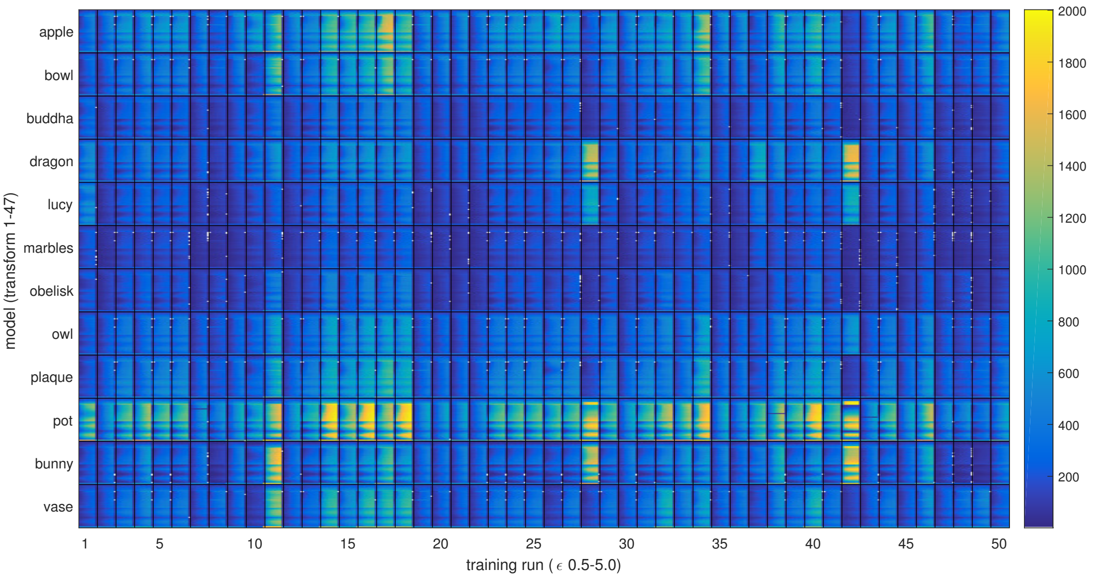


Figure D.9: Analysis F: Heatmap of TP repeated points for 3D training.

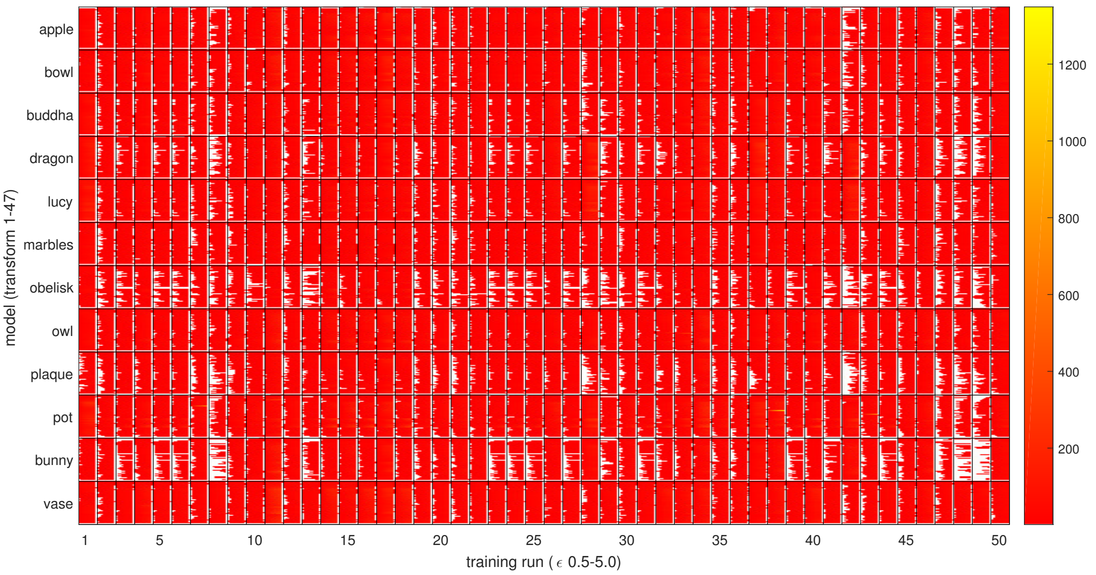


Figure D.10: Analysis F: Heatmap of FP repeated points for 3D training.

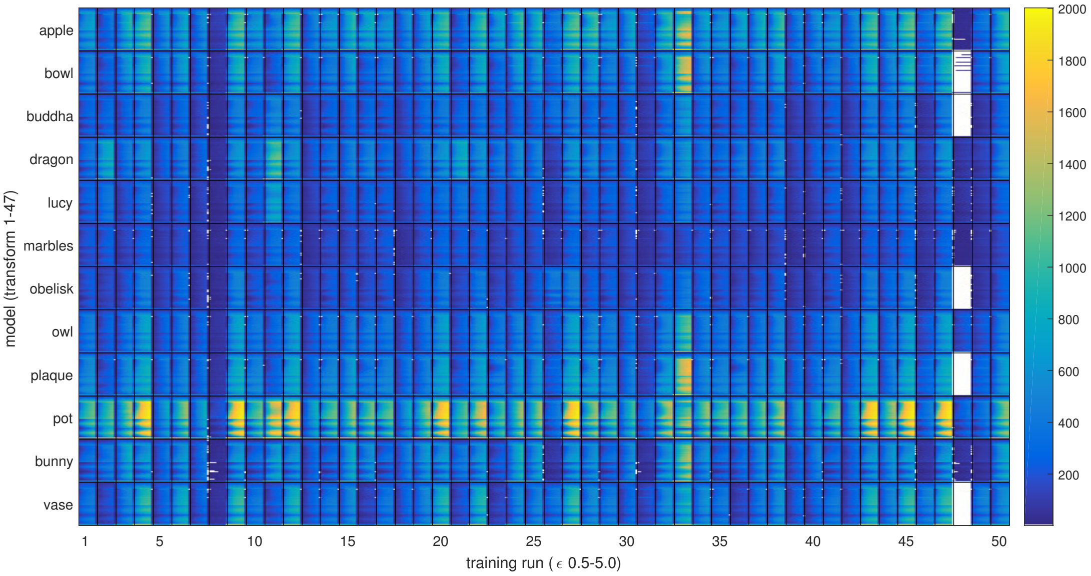


Figure D.11: Analysis F: Heatmap of TP repeated points for 2D training.

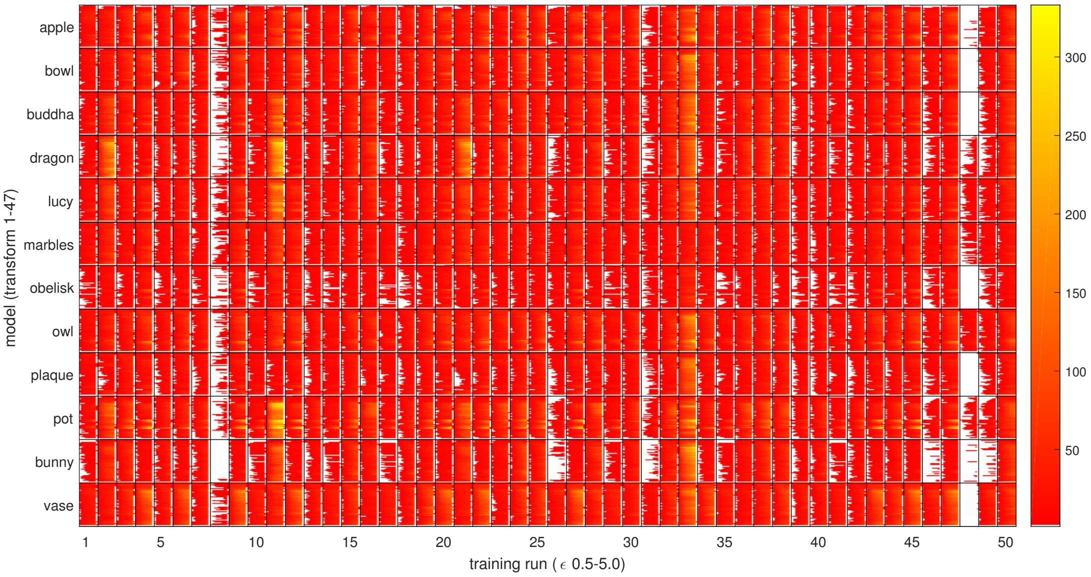


Figure D.12: Analysis F: Heatmap of FP repeated points for 2D training.

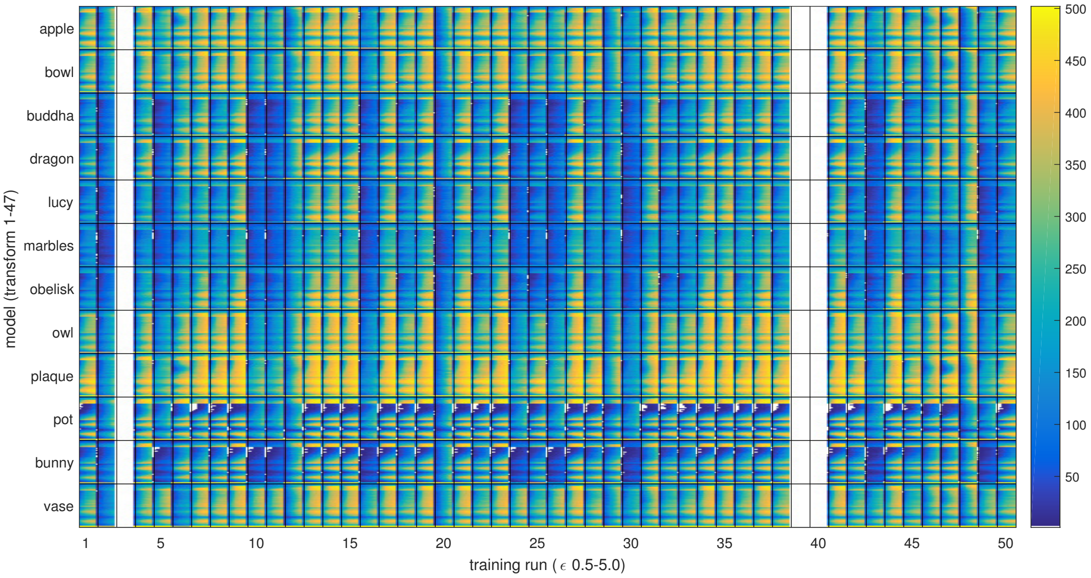


Figure D.13: Analysis G: Heatmap of TP repeated points for 3D training.

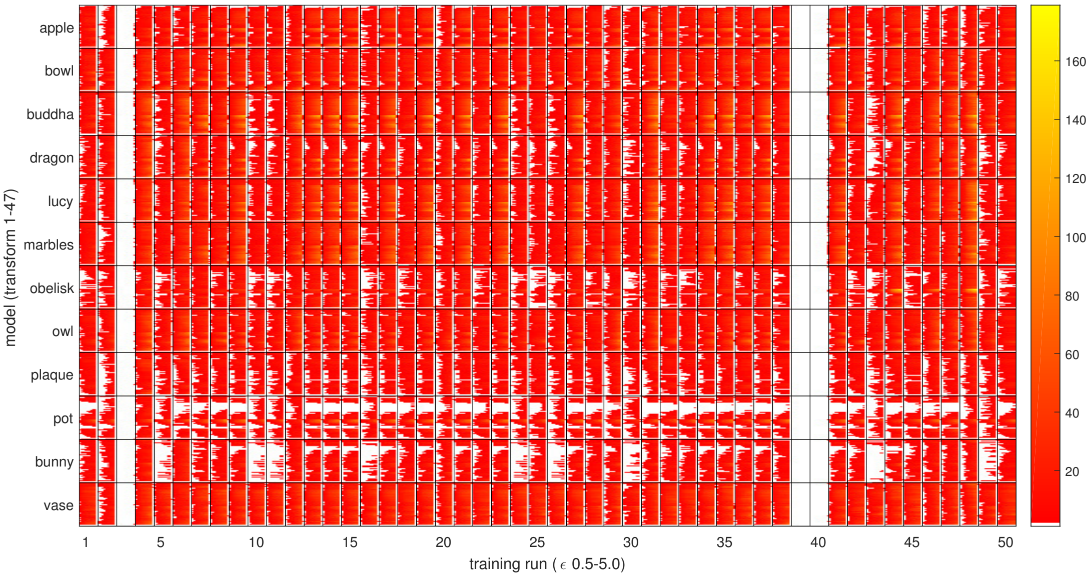


Figure D.14: Analysis G: Heatmap of FP repeated points for 3D training.

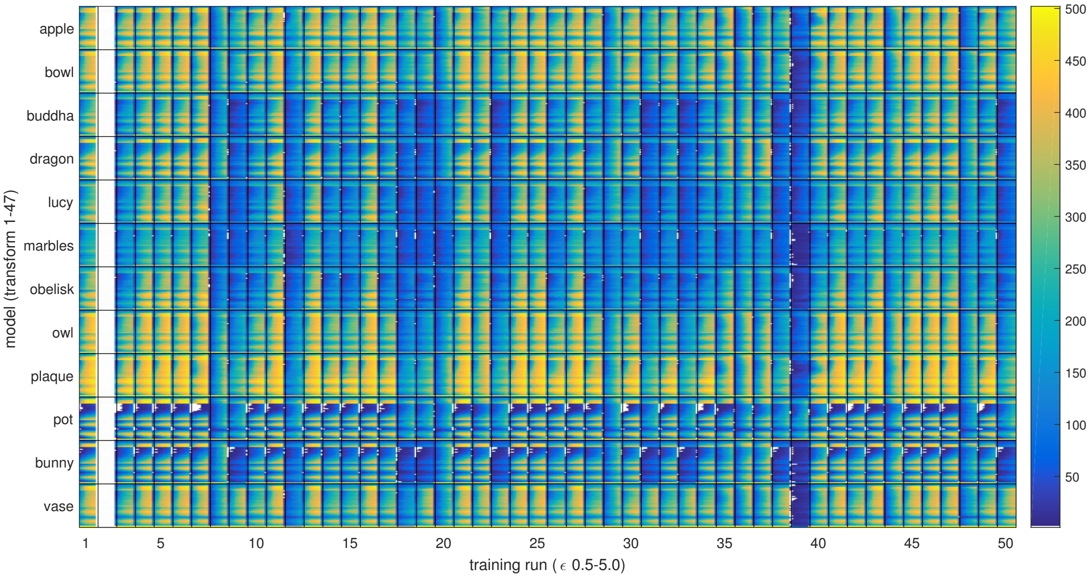


Figure D.15: Analysis G: Heatmap of TP repeated points for 2D training.

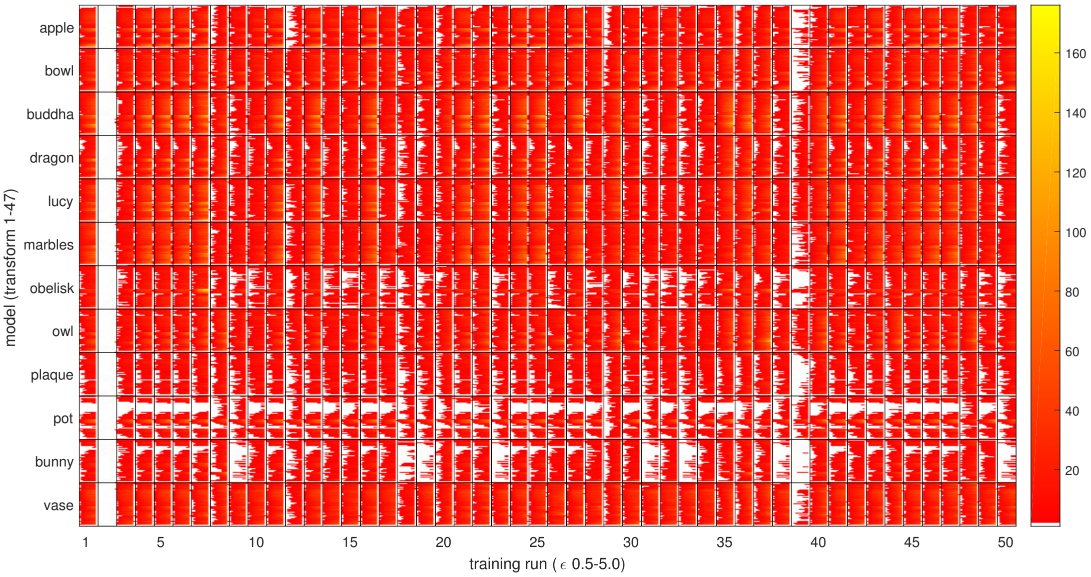


Figure D.16: Analysis G: Heatmap of FP repeated points for 2D training.

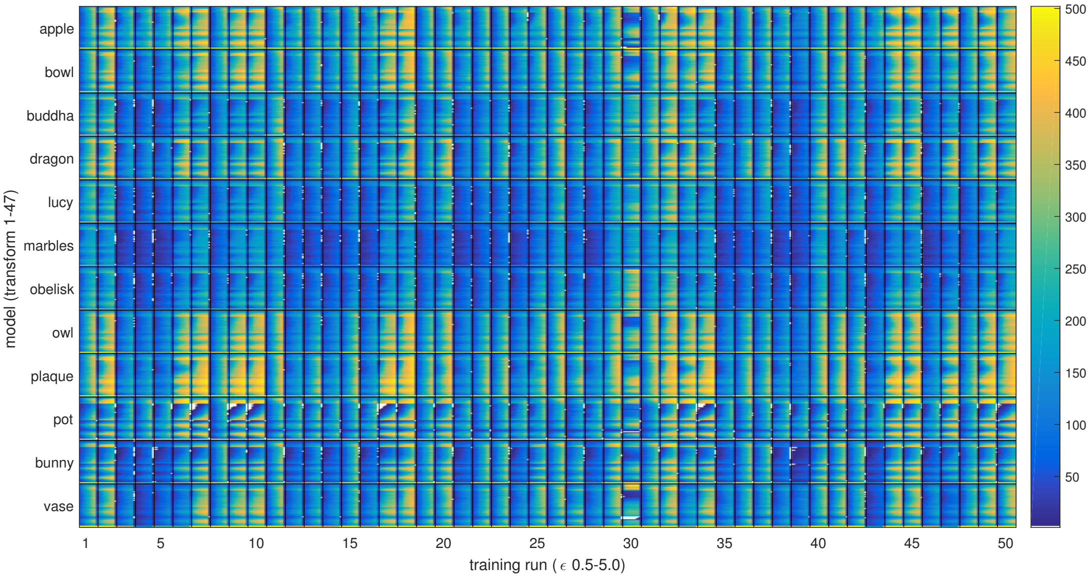


Figure D.17: Analysis H: Heatmap of TP repeated points for 3D training.

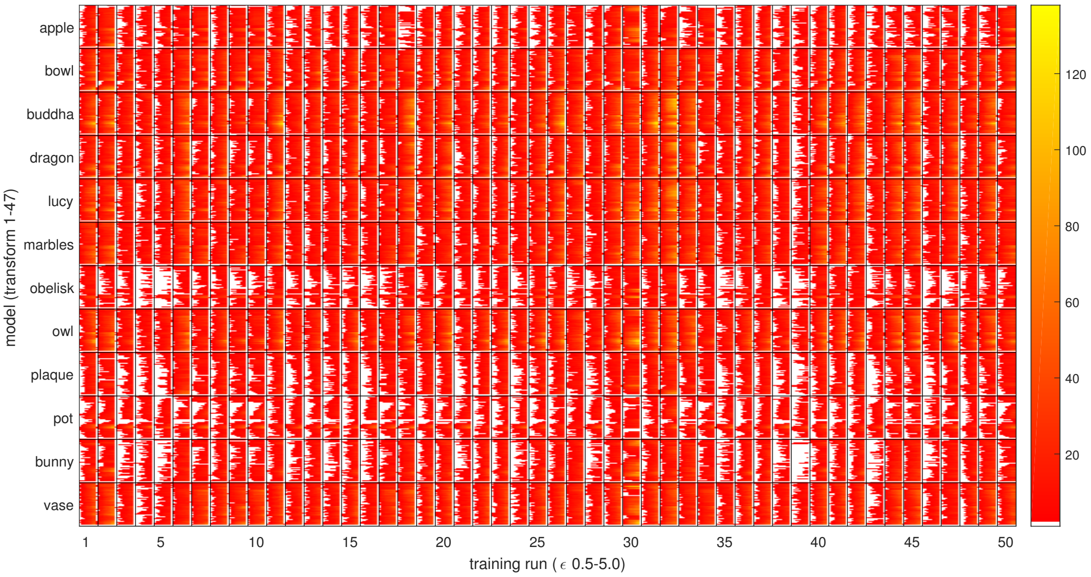


Figure D.18: Analysis H: Heatmap of FP repeated points for 3D training.

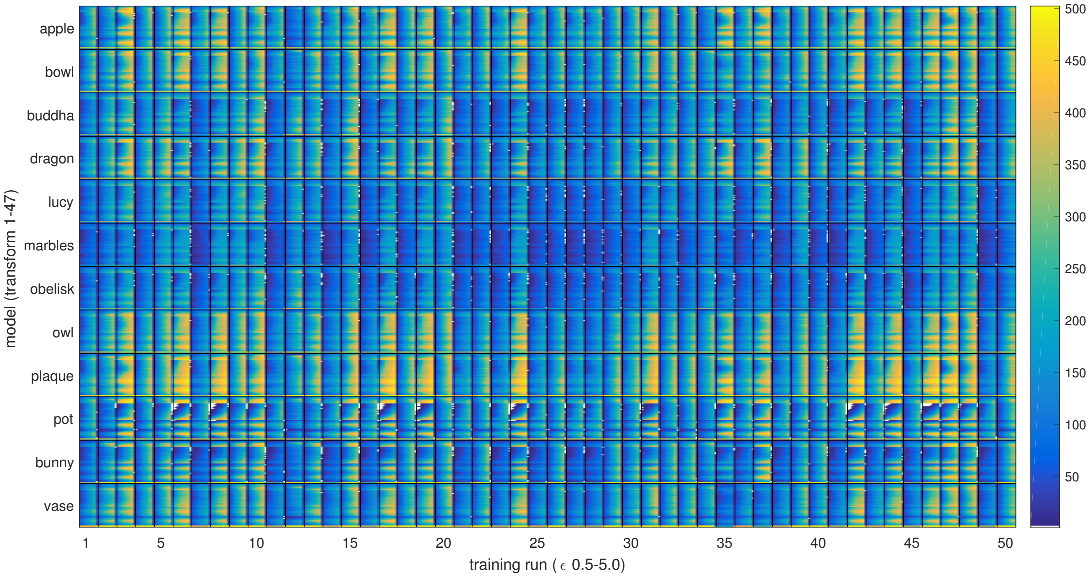


Figure D.19: Analysis H: Heatmap of TP repeated points for 2D training.

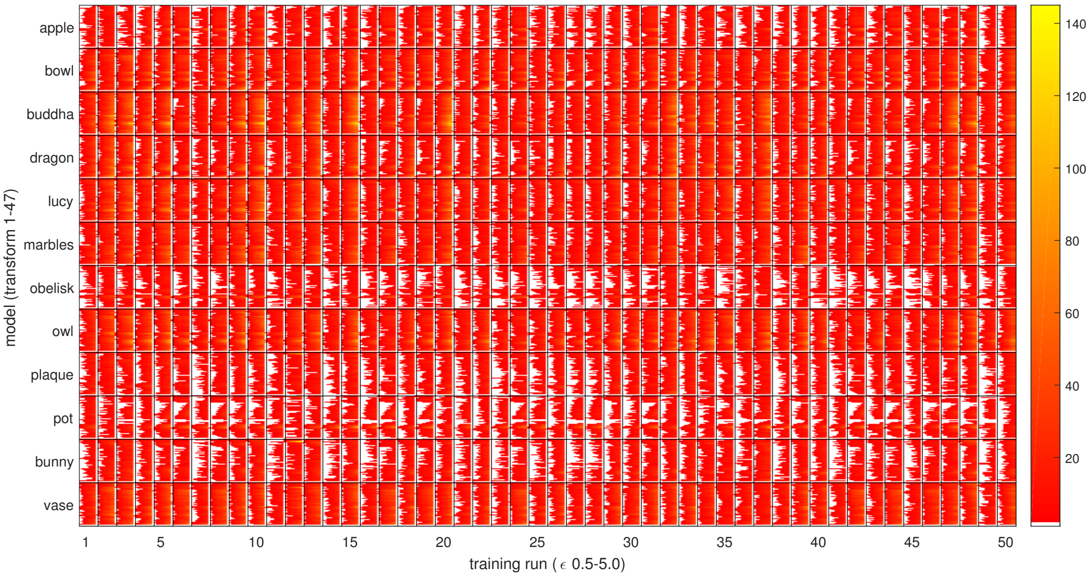


Figure D.20: Analysis H: Heatmap of FP repeated points for 2D training.

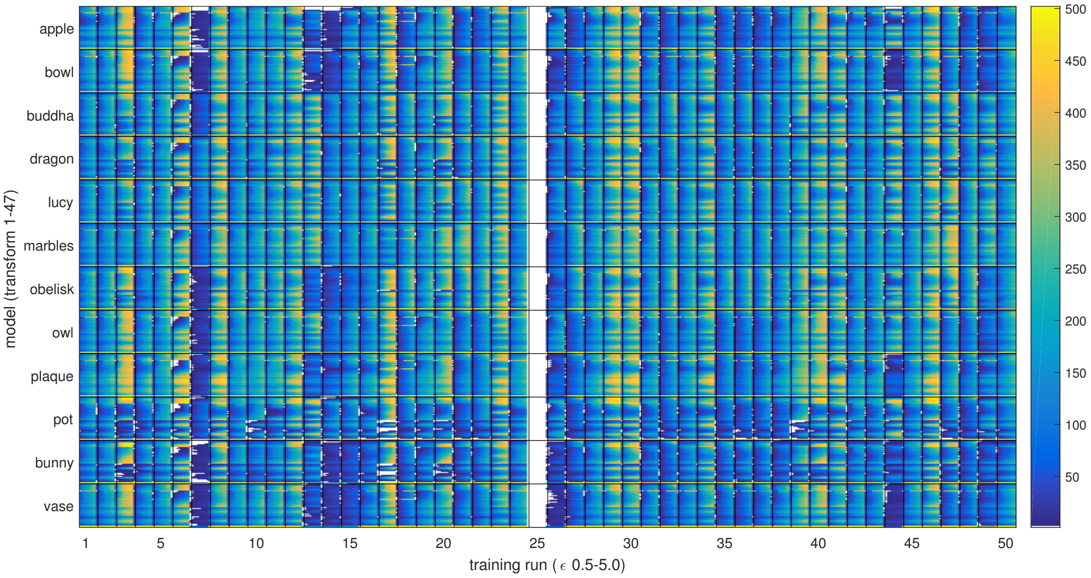


Figure D.21: Analysis I: Heatmap of TP repeated points for 3D training.

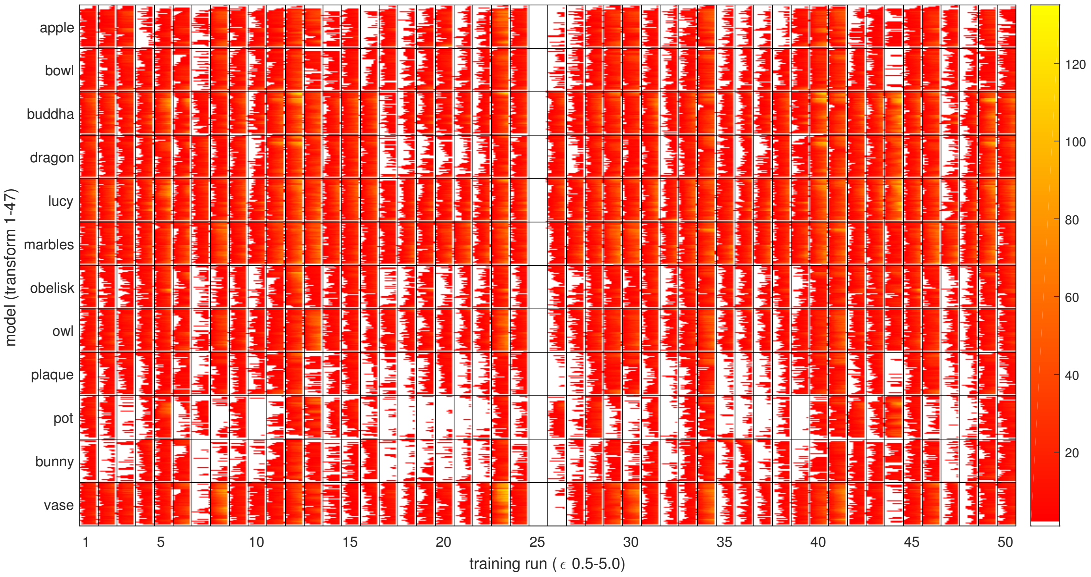


Figure D.22: Analysis I: Heatmap of FP repeated points for 3D training.

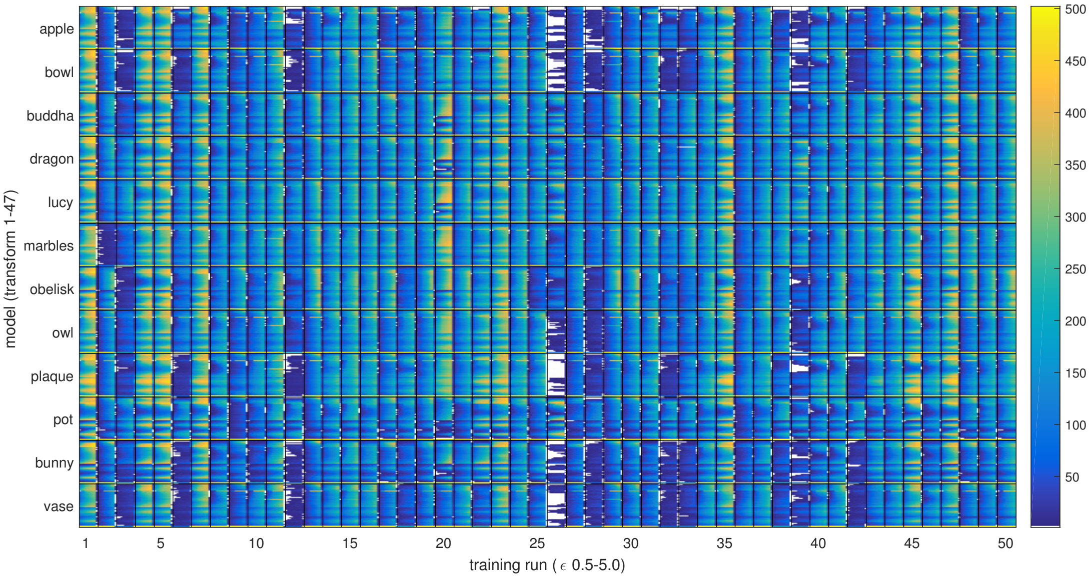


Figure D.23: Analysis I: Heatmap of TP repeated points for 3D training.

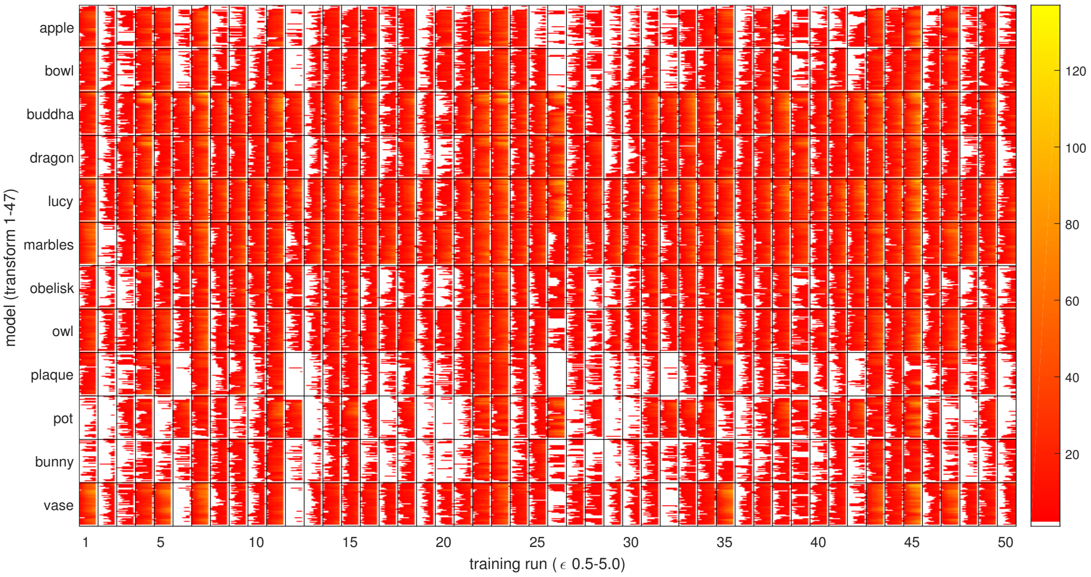


Figure D.24: Analysis I: Heatmap of FP repeated points for 3D training.

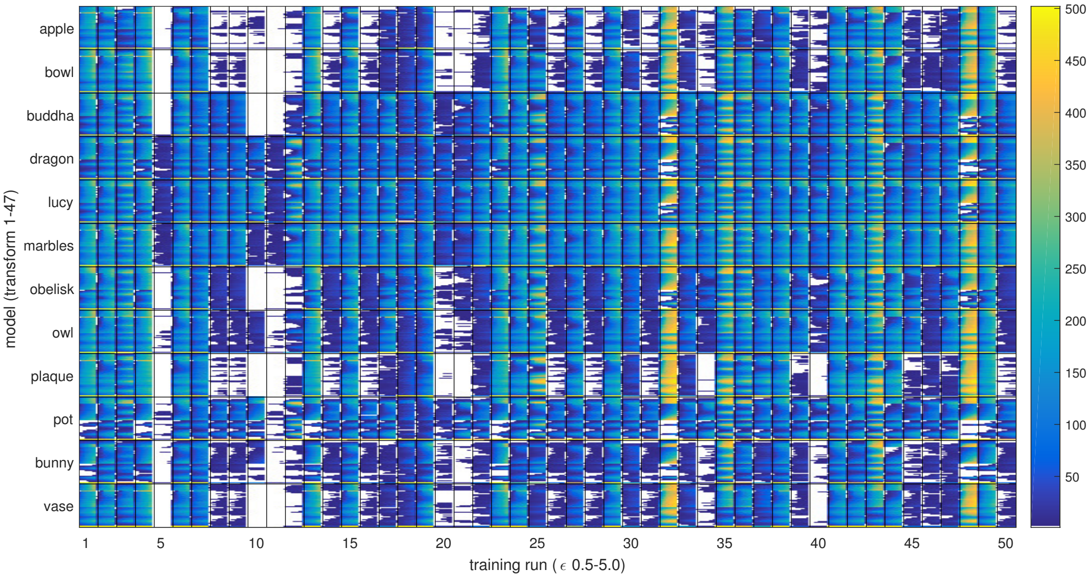


Figure D.25: Analysis J: Heatmap of TP repeated points for 3D training.

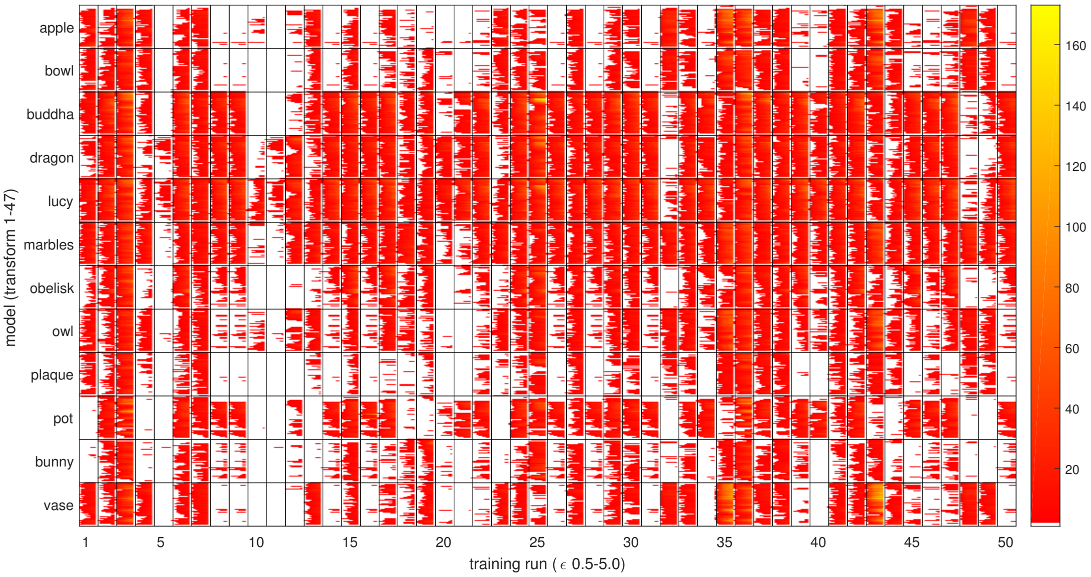


Figure D.26: Analysis J: Heatmap of FP repeated points for 3D training.

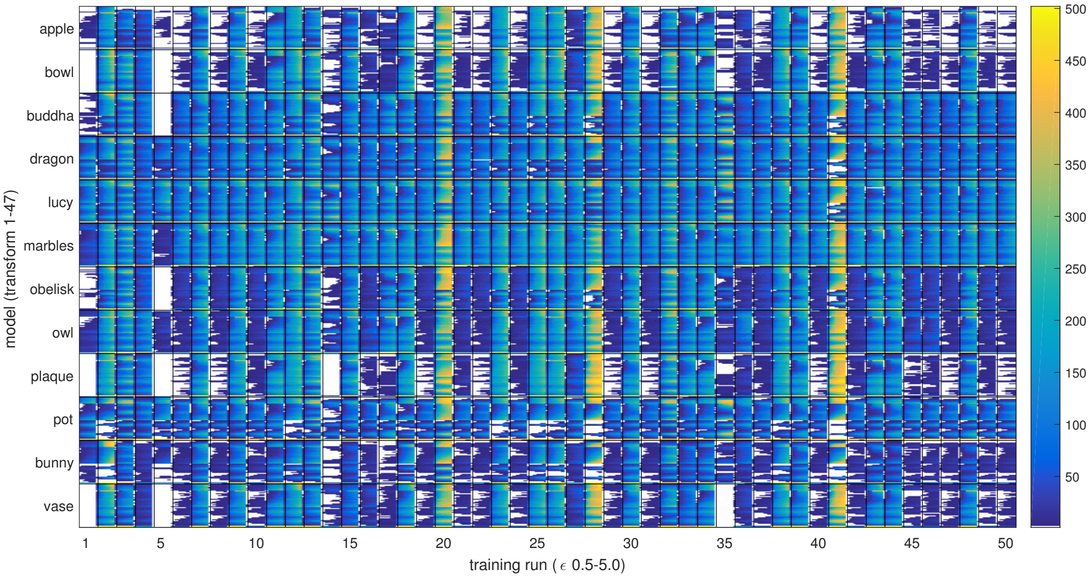


Figure D.27: Analysis J: Heatmap of TP repeated points for 2D training.

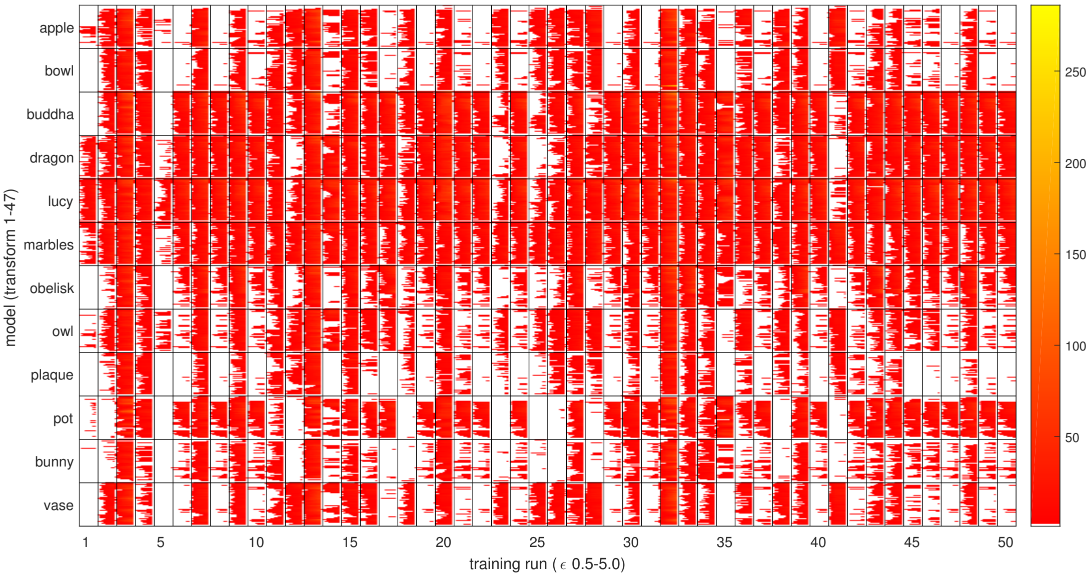


Figure D.28: Analysis J: Heatmap of FP repeated points for 2D training.

Bibliography

Abend, K., Harley, T. & Kanal, L. (1965), 'Classification of binary random patterns', *IEEE Transactions on Information Theory* **11**(4), 538–544.

Agarwal, S., Awan, A. & Roth, D. (2004), 'Learning to detect objects in images via a sparse, part-based representation', *IEEE transactions on pattern analysis and machine intelligence* **26**(11), 1475–1490.

Agarwal, S. & Roth, D. (2002), Learning a sparse representation for object detection, *in* 'European conference on computer vision', Springer, pp. 113–127.

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M. & Szeliski, R. (2009), Building rome in a day, *in* 'Computer Vision, 2009 IEEE 12th International Conference on', IEEE, pp. 72–79.

Ahmed, S., Zhang, M., Peng, L. & Xue, B. (2014), Multiple feature construction for effective biomarker identification and classification using genetic programming, *in* 'Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation', ACM, pp. 249–256.

Al-Sahaf, H., Al-Sahaf, A., Xue, B., Johnston, M. & Zhang, M. (2017), 'Automatically evolving rotation-invariant texture image descriptors by genetic programming', *IEEE Transactions on Evolutionary Computation* **21**(1), 83–101.

Alahi, A., Ortiz, R. & Vandergheynst, P. (2012), Freak: Fast retina keypoint, *in* '2012 IEEE Conference on Computer Vision and Pattern Recognition', Ieee, pp. 510–517.

Alcantarilla, P. F., Bartoli, A. & Davison, A. J. (2012), Kaze features, *in* 'European Conference on Computer Vision', Springer, pp. 214–227.

Angeline, P. J. (1996), 'An investigation into the sensitivity of genetic programming to the frequency of leaf selection during subtree crossover', *GECCO '96: Proceedings of the First Annual Conference on Genetic Programming* **1**, 21–29. GP-96 multiple types of mutation Sunspot Numbers data from http://www.ngdc.noaa.gov/stp/SOLAR/SSN/ssn.html.

Artieda, J., Sebastian, J. M., Campoy, P., Correa, J. F., Mondragón, I. F., Martínez, C. & Olivares, M. (2009), 'Visual 3-d slam from uavs', *Journal of Intelligent and Robotic Systems* **55**(4-5), 299.

Atyabi, A., Luerssen, M., Fitzgibbon, S. & Powers, D. M. (2012), Evolutionary feature selection and electrode reduction for eeg classification, *in* 'Evolutionary Computation (CEC), 2012 IEEE Congress on', IEEE, pp. 1–8.

Aussedat, C., Venail, F., Nguyen, Y., Lescanne, E., Marx, M. & Bakhos, D. (2017), 'Usefulness of temporal bone prototype for drilling training: A prospective study', *Clinical Otolaryngology* **42**(6), 1200–1205.

Awad, A. I. & Hassaballah, M. (2016), *Image Feature Detectors and Descriptors*, Springer.

Azad, P., Asfour, T. & Dillmann, R. (2009), Combining harris interest points and the sift descriptor for fast scale-invariant object recognition, *in* 'IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems', IEEE Press, Piscataway, NJ, USA, pp. 4275–4280.

Bäck, T. (1996), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press,, New York.

Bäck, T., Fogel, D. B. & Michalewicz, Z. (1997), *Handbook of evolutionary computation*, CRC Press.

Bäck, T., Graaf, J. M. d., Kok, J. N. & Kosters, W. A. (2001), 'Theory of genetic algorithms', *Current Trends in Theoretical Computer Science* **1**, 546–578.

Bäck, T., Rudolph, G. & Schwefel, H. (1993), Evolutionary programming and evolution strategies: Similarities and differences, *in* 'In Proceedings of the Second Annual Conference on Evolutionary Programming', pp. 11–22.

Bajcsy, R. (1973), 'Computer identification of visual surfaces', *Computer Graphics and Image Processing* **2**(2), 118 – 130.

Baluška, F., Lev-Yadun, S. & Mancuso, S. (2010), 'Swarm intelligence in plant roots', *Trends in ecology & evolution* **25**(12), 682–683.

Bandyopadhyay, S. & Maulik, U. (2002), 'Genetic clustering for automatic evolution of clusters and application to image classification', *Pattern recognition* **35**(6), 1197–1208.

Baumberg, A. (2000), Reliable feature matching across widely separated views, *in* 'Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on', Vol. 1, IEEE, pp. 774–781.

Beaudet, P. (1978), Rotationally invariant image operators, *in* 'Proc. Intl. Joint Conf. on Pattern Recognition', pp. 579–583.

Begley, M. R. (2017), 'Virtual simulation and design of barrier coatings for ceramic composites'.

Beni, G. & Wang, J. (1993), Swarm intelligence in cellular robotic systems, *in* 'Robots and Biological Systems: Towards a New Bionics?', Springer, pp. 703–712.

Bhanu, B., Lin, Y. & Krawiec, K. (2005), *Evolutionary synthesis of pattern recognition systems*, Springer.

Bhowan, U., Johnston, M. & Zhang, M. (2012), 'Developing new fitness functions in genetic programming for classification with unbalanced data', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **42**(2), 406–421.

Bi, Y., Zhang, M. & Xue, B. (2018), Genetic programming for automatic global and local feature extraction to image classification, *in* '2018 IEEE Congress on Evolutionary Computation (CEC)', IEEE, pp. 1–8.

Bishop, C. M. (2006), 'Pattern recognition and machine learning (information science and statistics) springer-verlag new york', *Inc. Secaucus, NJ, USA* .

Blake, A. & Isard, M. (1998), *Active Contours: The Application of Techniques from Graphics,Vision,Control Theory and Statistics to Visual Tracking of Shapes in Motion*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Bohm, W. & Geyer-Schulz, A. (1996), Exact uniform initialization for genetic programming, *in* R. K. Belew & M. Vose, eds, 'Foundations of Genetic

Algorithms IV', Morgan Kaufmann, University of San Diego, CA, USA, pp. 379–407.

Bonabeau, E., Marco, D. d. R. D. F., Dorigo, M., Théraulaz, G., Theraulaz, G. et al. (1999), *Swarm intelligence: from natural to artificial systems*, number 1, Oxford university press.

Bowyer, K., Kranenburg, C. & Dougherty, S. (2001), 'Edge detector evaluation using empirical roc curves', *Computer Vision and Image Understanding* **84**(1), 77–103.

Bradley, A. P. (1997), 'The use of the area under the roc curve in the evaluation of machine learning algorithms', *Pattern recognition* **30**(7), 1145–1159.

Brameier, M. & Banzhaf, W. (2001), 'A comparison of linear genetic programming and neural networks in medical data mining', *IEEE Transactions on Evolutionary Computation* **5**(1), 17–26.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32.

Brice, C. R. & Fennema, C. L. (1970), 'Scene analysis using regions', *Artificial Intelligence* **1**(3-4), 205 – 226.

Bundy, A. & Wallen, L. (1984), Difference of gaussians, *in* 'Catalogue of Artificial Intelligence Tools', Springer, pp. 30–30.

Burt, P. J. & Adelson, E. H. (1987), The laplacian pyramid as a compact image code, *in* 'Readings in Computer Vision', Elsevier, pp. 671–679.

Cagnoni, S. (2014), Evolutionary image analysis and signal processing, *in* 'Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation', ACM, pp. 795–818.

Cagnoni, S., Dobrzeniecki, A. B., Poli, R. & Yanch, J. C. (1999), 'Genetic algorithm-based interactive segmentation of 3d medical images', *Image and Vision Computing* **17**(12), 881–895.

Cagnoni, S., Mordonini, M. & Sartori, J. (2007), Particle swarm optimization for object detection and segmentation, *in* 'Workshops on Applications of Evolutionary Computation', Springer, pp. 241–250.

Canny, J. (1986), 'A computational approach to edge detection', *IEEE Transactions on pattern analysis and machine intelligence* (6), 679–698.

Castro, L. N., De Castro, L. N. & Timmis, J. (2002), *Artificial immune systems: a new computational intelligence approach*, Springer Science & Business Media.

Chandrashekar, G. & Sahin, F. (2014), 'A survey on feature selection methods', *Computers & Electrical Engineering* **40**(1), 16–28.

Chen, H. & Bhanu, B. (2007), '3d free-form object recognition in range images using local surface patches', *Pattern Recognition Letters* **28**(10), 1252–1262.

Chen, H.-Y., Huang, C.-L. & Fu, C.-M. (2008), 'Hybrid-boost learning for multi-pose face detection and facial expression recognition', *Pattern Recogn.* **41**(3), 1173–1185. 1298940.

Chen, J., Shan, S., He, C., Zhao, G., Pietikainen, M., Chen, X. & Gao, W. (2010), 'Wld: A robust local image descriptor', *IEEE transactions on pattern analysis and machine intelligence* **32**(9), 1705–1720.

Cheng, H.-D., Cai, X., Chen, X., Hu, L. & Lou, X. (2003), 'Computer-aided detection and classification of microcalcifications in mammograms: a survey', *Pattern recognition* **36**(12), 2967–2991.

Chinchor, N. (1992), Muc-4 evaluation metrics, *in* 'Proceedings of the Fourth Message Understanding Conference', p. pp. 2229.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A. et al. (2007), *Evolutionary algorithms for solving multi-objective problems*, Vol. 5, Springer.

Cohen, A. R., Lohani, S., Manjila, S., Natsupakpong, S., Brown, N. & Cavusoglu, M. C. (2013), 'Virtual reality simulation: basic concepts and use in endoscopic neurosurgery training', *Child's Nervous System* **29**(8), 1235–1244.

Darwin, C. (1872), 'On the origin of species by means of natural selection', *Journal of the Proceedings of the Linnean Society* **1**, 502.

Dawkins, R. (1983), *Evolution from Molecules to Man*, Cambridge University Press.

Dawn, D. D. & Shaikh, S. H. (2016), 'A comprehensive survey of human action recognition with spatio-temporal interest point (stip) detector', *The Visual Computer* **32**(3), 289–306.

D'backhaeseleer, P. (1994), Context preserving crossover in genetic programming, *in* 'Proceedings of the 1994 IEEE World Congress on Computational Intelligence', Vol. 1, IEEE Press, Orlando, Florida, USA, pp. 256–261.

Deng, H., Zhang, W., Mortensen, E., Dietterich, T. & Shapiro, L. (2007), Principal curvature-based region detector for object recognition, *in* 'Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on', IEEE, pp. 1–8.

Dignum, S. & Poli, R. (2007), Generalisation of the limiting distribution of program sizes in tree-based genetic programming and analysis of its effects on bloat, *in* D. Thierens, H.-G. Beyer, J. Bongard, J. Branke, J. A. Clark, D. Cliff, C. B. Congdon, K. Deb, B. Doerr, T. Kovacs, S. Kumar, J. F. Miller, J. Moore, F. Neumann, M. Pelikan, R. Poli, K. Sastry, K. O. Stanley, T. Stutzle, R. A. Watson & I. Wegener, eds, 'GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation', Vol. 2, ACM Press, London, pp. 1588–1595. GECCO-2007 A joint meeting of the sixteenth international conference on genetic algorithms (ICGA-2007) and the twelfth annual genetic programming conference (GP-2007). ACM Order Number 910071.

Dominy, N. J., Mills, S. T., Yakacki, C. M., Roscoe, P. B. & Carpenter, R. D. (2018), 'New guinea bone daggers were engineered to preserve social prestige', *Royal Society open science* **5**(4), 172067.

Dorigo, M., Maniezzo, V. & Colorni, A. (1991), 'Positive feedback as a search strategy'.

Dorigo, M., Maniezzo, V. & Colorni, A. (1996), 'Ant system: optimization by a colony of cooperating agents', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **26**(1), 29–41.

E. Rosten, T. D. (2006), Machine learning for high-speed corner detection, *in* 'European Conference on Computer Vision'.

Eberhart, R. & Kennedy, J. (1995), A new optimizer using particle swarm theory, *in* 'Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on', IEEE, pp. 39–43.

Eberly, D., Gardner, R., Morse, B., Pizer, S. & Scharlach, C. (1994), 'Ridges for image analysis', *Journal of Mathematical Imaging and Vision* **4**, 353–373.

Ebner, M. (1997), On the evolution of edge detectors for robot vision using genetic programming, *in* 'Workshop SOAVE', Vol. 97, pp. 127–134.

Ebner, M., karls-universitat Tubingen, E. & Rechnerarchitektur, A. (1998), On the evolution of interest operators using genetic programming, *in* 'In Proc. EuroGP98', pp. 6–10.

Ebner, M. et al. (1998), 'On the evolution of interest operators using genetic programming', *COGNITIVE SCIENCE RESEARCH PAPERS-UNIVERSITY OF BIRMINGHAM CSRP* pp. 6–10.

Eklund, S. E. (2002), A massively parallel GP engine in VLSI, *in* D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow & M. Shackleton, eds, 'Proceedings of the 2002 Congress on Evolutionary Computation CEC2002', IEEE Press, pp. 629–633. CEC 2002 - A joint meeting of the IEEE, the Evolutionary Programming Society, and the IEE. Held in connection with the World Congress on Computational Intelligence (WCCI 2002).

El Ferchichi, S., Zidi, S., Laabidi, K., Ksouri, M. & Maouche, S. (2011), A new feature extraction method based on clustering for face recognition, *in* 'Engineering Applications of Neural Networks', Springer, pp. 247–253.

Entwisle, J. & Powers, D. M. (1998), The present use of statistics in the evaluation of nlp parsers, *in* 'Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning', Association for Computational Linguistics, pp. 215–224.

Everts, I., Van Gemert, J. C. & Gevers, T. (2014), 'Evaluation of color spatio-temporal interest points for human action recognition', *IEEE Transactions on Image Processing* **23**(4), 1569–1580.

Evolving Computer Programs without Subtree Crossover (1997), Vol. 1.

Fan, H., Su, H. & Guibas, L. J. (2017), A point set generation network for 3d object reconstruction from a single image., *in* 'CVPR', Vol. 2, p. 6.

Fawcett, T. (2004), 'Roc graphs: Notes and practical considerations for researchers', *Machine learning* **31**(1), 1–38.

Fernández-Palacios, B. J., Morabito, D. & Remondino, F. (2017), 'Access to complex reality-based 3d models using virtual reality solutions', *Journal of cultural heritage* **23**, 40–48.

Flach, P. A. (2003), The geometry of roc space: understanding machine learning metrics through roc isometrics, *in* 'Proceedings of the 20th International Conference on Machine Learning (ICML-03)', pp. 194–201.

Flint, A., Dick, A. & Van Den Hengel, A. (2007), Thrift: Local 3d structure recognition, *in* 'dicta', IEEE, pp. 182–188.

Flint, A., Dick, A. & Van den Hengel, A. (2008), 'Local 3d structure recognition in range images', *IET Computer Vision* **2**(4), 208–217.

Forssén, P.-E. (2007), Maximally stable colour regions for recognition and matching, *in* 'Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on', IEEE, pp. 1–8.

Förstner, W. (1986), A feature based correspondence algorithms for image matching, *in* 'Intl. Arch. Photogrammetry and Remote Sensing', Vol. 24, pp. 160–166.

Freund, Y. & Schapire, R. E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of computer and system sciences* **55**(1), 119–139.

Frome, A., Huber, D., Kolluri, R., Bülow, T. & Malik, J. (2004), Recognizing objects in range data using regional point descriptors, *in* 'European conference on computer vision', Springer, pp. 224–237.

Fu, W., Johnston, M. & Zhang, M. (2011), Genetic programming for edge detection: a global approach, *in* 'Evolutionary Computation (CEC), 2011 IEEE Congress on', IEEE, pp. 254–261.

Fu, W., Johnston, M. & Zhang, M. (2013), Automatic construction of gaussian-based edge detectors using genetic programming, *in* 'European Conference on the Applications of Evolutionary Computation', Springer, pp. 365–375.

Fu, W., Johnston, M. & Zhang, M. (2016), 'Genetic programming for edge detection: a gaussian-based approach', *Soft Computing* **20**(3), 1231–1248.

G., O. & L., T. (2011), 'Evolutionary computer assisted design of image operators that detect interest points using genetic programming.', *Image and Vision Computing. Elsevier.* **29**, 484–498.

Gabriel, P., Hayet, J.-B., Piater, J. & Verly, J. (2005), Object tracking using color interest points, *in* 'Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on', IEEE, pp. 159–164.

Gao, H., Xu, W., Sun, J. & Tang, Y. (2010), 'Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm', *IEEE Transactions on Instrumentation and Measurement* **59**(4), 934–946.

Gauglitz, S., Höllerer, T. & Turk, M. (2011), 'Evaluation of interest point detectors and feature descriptors for visual tracking', *International journal of computer vision* **94**(3), 335.

Gauglitz, S., Hllerer, T. & Turk, M. (2011), 'Evaluation of interest point detectors and feature descriptors for visual tracking', *International Journal of Computer Vision* **94**, 335–360.

Gil, A., Mozos, O. M., Ballesta, M. & Reinoso, O. (2010), 'A comparative evaluation of interest point detectors and local descriptors for visual slam', *Machine Vision and Applications* **21**(6), 905–920.

Gomez, C. H., Medathati, K., Kornprobst, P., Murino, V. & Sona, D. (2015), Improving freak descriptor for image classification, *in* 'International Conference on Computer Vision Systems', Springer, pp. 14–23.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016), *Deep learning*, Vol. 1, MIT press Cambridge.

Grauman, K. & Leibe, B. (2011), 'Visual object recognition', *Synthesis lectures on artificial intelligence and machine learning* **5**(2), 1–181.

Guiducci, A. (1988), 'Corner characterization by differential geometry techniques', *Pattern Recogn. Lett.* **8**(5), 311–318.

Guillaume Gals, Alain Crouzil, S. C. (2010), Complementarity of feature point detectors, *in* 'In International Joint Conference on Computer Vision Theory and Applications'.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J. & Kwok, N. M. (2016*a*), 'A comprehensive performance evaluation of 3d local feature descriptors', *International Journal of Computer Vision* **116**(1), 66–89.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J. & Kwok, N. M. (2016*b*), 'A comprehensive performance evaluation of 3d local feature descriptors', *International Journal of Computer Vision* **116**(1), 66–89.
URL: *https://doi.org/10.1007/s11263-015-0824-y*

Guo, Y., Sohel, F. A., Bennamoun, M., Lu, M. & Wan, J. (2013*a*), Trisi: A distinctive local surface descriptor for 3d modeling and object recognition., *in* 'GRAPP/IVAPP', pp. 86–93.

Guo, Y., Sohel, F., Bennamoun, M., Lu, M. & Wan, J. (2013*b*), 'Rotational projection statistics for 3d local surface description and object recognition', *International journal of computer vision* **105**(1), 63–86.

Gupta, N., Gupta, R., Singh, A. & Wytock, M. (2008), 'Object recognition using template matching', *Available in: https://tmatch. googlecode. com/svnhistory/r38/trunk/report/report. pdf* .

Gustavo, O. & Leonardo, T. (2006), 'Using evolution to learn how to perform interest point detection', *Pattern Recognition, International Conference on* **1**, 211–214.

Hand, D. J. (2009), 'Measuring classifier performance: a coherent alternative to the area under the roc curve', *Machine learning* **77**(1), 103–123.

Hänsch, R., Weber, T. & Hellwich, O. (2014), 'Comparison of 3d interest point detectors and descriptors for point cloud fusion', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2**(3), 57.

Haralick, R. M., Shanmugam, K., Dinstein, I. et al. (1973), 'Textural features for image classification', *IEEE Transactions on systems, man, and cybernetics* **3**(6), 610–621.

Harries, K. & Smith, P. (1997), 'Exploring alternative operators and search strategies in geneticprogramming', pp. 147–155.

Harris, B. T., Montero, D., Grant, G. T., Morton, D., Llop, D. R. & Lin, W.-S. (2017), 'Creation of a 3-dimensional virtual dental patient for computer-guided surgery and cad-cam interim complete removable and fixed dental prostheses: a clinical report', *The Journal of prosthetic dentistry* **117**(2), 197–204.

Harris, C. & Stephens, M. (1988), A combined corner and edge detector, *in* 'Alvey Vision Conference', p. 147151.

Hartley, R. & Zisserman, A. (2003), *Multiple View Geometry in Computer Vision*, Cambridge University Press, New York, NY, USA.

Herbert Bay, Tinne Tuytelaars, L. V. G. (2006), Surf: Speeded up robust features, *in* 'Lecture Notes in Computer Science', Vol. 3951, pp. 404–417.

Herrmann, M., Mayer, C. & Radig, B. (2014), 'Automatic generation of image analysis programs', *Pattern recognition and image analysis* **24**(3), 400–408.

Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006), 'A fast learning algorithm for deep belief nets', *Neural computation* **18**(7), 1527–1554.

Hiremath, P. & Pujari, J. (2008), 'Content based image retrieval using color boosted salient points and shape features of an image', *International Journal of Image Processing* **2**(1), 10–17.

Holland, J. H. (1992), *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, USA.

Holland, J. H., Booker, L. B., Colombetti, M., Dorigo, M., Goldberg, D. E., Forrest, S., Riolo, R. L., Smith, R. E., Lanzi, P. L., Stolzmann, W. et al. (1999), What is a learning classifier system?, *in* 'International Workshop on Learning Classifier Systems', Springer, pp. 3–32.

Iba, H. (1996), Random tree generation for genetic programming, *in* H.-M. Voigt, W. Ebeling, I. Rechenberg & H.-P. Schwefel, eds, 'Parallel Problem Solving from Nature IV, Proceedings of the International Conference on Evolutionary Computation', Vol. 1141 of *LNCS*, Springer Verlag, Berlin, Germany, pp. 144–153.

Jang, S., Vitale, J. M., Jyung, R. W. & Black, J. B. (2017), 'Direct manipulation is better than passive viewing for learning anatomy in a three-dimensional virtual reality environment', *Computers & Education* **106**, 150–165.

John, G. H., Kohavi, R. & Pfleger, K. (1994), Irrelevant features and the subset selection problem, *in* 'Machine Learning Proceedings 1994', Elsevier, pp. 121–129.

Johnson, A. E. & Hebert, M. (1999), 'Using spin images for efficient object recognition in cluttered 3d scenes', *IEEE Transactions on Pattern Analysis & Machine Intelligence* (5), 433–449.

Kanade, C. T. T. (April 1991), . detection and tracking of point features, *in* 'Carnegie Mellon University Technical Report CMU-CS-91-132'.

Karaboga, D. & Akay, B. (2009), 'A survey: algorithms simulating bee swarm intelligence', *Artificial intelligence review* **31**(1-4), 61.

Karaboga, D. & Basturk, B. (2008), ‘On the performance of artificial bee colony (abc) algorithm’, *Applied soft computing* **8**(1), 687–697.

Karaboga, D. & Ozturk, C. (2011), ‘A novel clustering approach: Artificial bee colony (abc) algorithm’, *Applied soft computing* **11**(1), 652–657.

Käthe, U. (2000), Generische Programmierung fr die Bildverarbeitung, PhD thesis, University of Hamburg.

Ke, Y. & Sukthankar, R. (2004), Pca-sift: A more distinctive representation for local image descriptors, *in* ‘Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on’, Vol. 2, IEEE, pp. II–II.

Kennedy, R. (1995), J. and eberhart, particle swarm optimization, *in* ‘Proceedings of IEEE International Conference on Neural Networks IV, pages’, Vol. 1000.

Kim, J.-H., Park, Y.-C., Yu, H.-S., Kim, M.-K., Kang, S.-H. & Choi, Y. J. (2017), ‘Accuracy of 3-dimensional virtual surgical simulation combined with digital teeth alignment: A pilot study’, *Journal of Oral and Maxillofacial Surgery* **75**(11), 2441–e1.

Kinnear, Jr., K. E. (1993), Evolving a sort: Lessons in genetic programming.

Koller, D. & Sahami, M. (1996), Toward optimal feature selection, Technical report, Stanford InfoLab.

Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA.

Koza, J. R. (1994), *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, Cambridge Massachusetts.

Kraft, D. H., Petry, F. E., Buckles, W. P. & Sadasivan, T. (1994), The use of genetic programming to build queries for information retrieval, *in* ‘Proceedings of the 1994 IEEE World Congress on Computational Intelligence’, IEEE Press, Orlando, Florida, USA, pp. 468–473.

Labatut, P., Pons, J.-P. & Keriven, R. (2007), Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts, *in* ‘Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on’, IEEE, pp. 1–8.

Lang, S. R., Luerssen, M. H. & Powers, D. M. (2013*a*), Automated evaluation of interest point detectors, *in* 'Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on', IEEE, pp. 443–447.

Lang, S. R., Luerssen, M. H. & Powers, D. M. W. (2013*b*), *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, Springer International Publishing, Heidelberg, chapter Repeatability Measurements for 2D Interest Point Detectors on 3D Models, pp. 361–370.

Lang, S. R., Luerssen, M. H. & Powers, D. M. W. (2014), 'Automated evaluation of interest point detectors', *Int. J. Softw. Innov.* **2**(1), 86–105.
URL: *http://dx.doi.org/10.4018/ijsi.2014010107*

Langdon, W. B. (1998), 'The evolution of size in variable length representations', pp. 633–638. ICEC-98 Held In Conjunction With WCCI-98 — 1998 IEEE World Congress on Computational Intelligence.

Langdon, W. B. (1999), *Size Fair and Homologous Tree Genetic Programming Crossovers*, Vol. 2, Morgan Kaufmann, Orlando, Florida, USA.

Langdon, W. B. (2000), 'Size fair and homologous tree genetic programming crossovers', *Genetic Programming and Evolvable Machines* **1**(1/2), 95–119.

Langdon, W. B. & Poli, R. (1998), Why ants are hard, *in* J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba & R. Riolo, eds, 'Genetic Programming 1998: Proceedings of the Third Annual Conference', Morgan Kaufmann, University of Wisconsin, Madison, Wisconsin, USA, pp. 193–201.

Le, Q. V. (2013), Building high-level features using large scale unsupervised learning, *in* 'Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on', IEEE, pp. 8595–8598.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989), 'Backpropagation applied to handwritten zip code recognition', *Neural computation* **1**(4), 541–551.

Lee, E. & Park, H. (2017), '3d virtual fit simulation technology: strengths and areas of improvement for increased industry adoption', *International Journal of Fashion Design, Technology and Education* **10**(1), 59–70.

Lensen, A., Al-Sahaf, H., Zhang, M. & Xue, B. (2015), A hybrid genetic programming approach to feature detection and image classification, *in* 'Image and Vision Computing New Zealand (IVCNZ), 2015 International Conference on', IEEE, pp. 1–6.

Lensen, A., Al-Sahaf, H., Zhang, M. & Xue, B. (2016), Genetic programming for region detection, feature extraction, feature construction and classification in image data, *in* 'European Conference on Genetic Programming', Springer, pp. 51–67.

Liang, Y., Zhang, M. & Browne, W. N. (2015), A supervised figure-ground segmentation method using genetic programming, *in* 'European Conference on the Applications of Evolutionary Computation', Springer, pp. 491–503.

Lin, X., Zhu, C., Zhang, Q., Huang, X. & Liu, Y. (2017), 'Efficient and robust corner detectors based on second-order difference of contour', *IEEE Signal Processing Letters* **24**(9), 1393–1397.

Lin, Y. & Bhanu, B. (2005), 'Evolutionary feature synthesis for object recognition', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **35**(2), 156–171.

Lindeberg, T. (1993), 'Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention', *International Journal of Computer Vision* **11**(3), 283–318.

Lindeberg, T. (1998), 'Edge detection and ridge detection with automatic scale selection', *International Journal of Computer Vision* **30**(2), 117–156.

Lindeberg, T. (2013), 'Scale selection properties of generalized scale-space interest point detectors', *Journal of Mathematical Imaging and vision* **46**(2), 177–210.

Lindeberg, T. (2015*a*), 'Image matching using generalized scale-space interest points', *Journal of Mathematical Imaging and Vision* **52**(1), 3–36.
URL: *https://doi.org/10.1007/s10851-014-0541-0*

Lindeberg, T. (2015*b*), 'Image matching using generalized scale-space interest points', *Journal of Mathematical Imaging and Vision* **52**(1), 3–36.

Liu, H. & Motoda, H. (1998), *Feature extraction, construction and selection: A data mining perspective*, Vol. 453, Springer Science & Business Media.

Liu, S.-T. & Tsai, W.-H. (1990), 'Moment-preserving corner detection', *Pattern Recognition* **23**(5), 441 – 460.

Loog, M. & Lauze, F. (2010), 'The improbability of harris interest points', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 1141–1147.

Lowe, D. G. (1999), Object recognition from local scale-invariant features, *in* 'Computer vision, 1999. The proceedings of the seventh IEEE international conference on', Vol. 2, Ieee, pp. 1150–1157.

Lowe, D. G. (2004), 'Distinctive image features from scale-invariant keypoints', *International Journal of Computer Vision* **60**, 91–110.

Luke, S. (2000), 'Two fast tree-creation algorithms for genetic programming', *IEEE Transactions on Evolutionary Computation* **4**(3), 274–283. URL `http://www.cs.gmu.edu/~sean/papers/treecreation.pdf`.

Lunscher, W. H. & Beddoes, M. P. (1986), 'Optimal edge detector design i: Parameter selection and noise effects', *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2), 164–177.

Malik, J., Belongie, S., Shi, J. & Leung, T. (1999), 'Textons, contours and regions: Cue integration in image segmentation', *Computer Vision, IEEE International Conference on* **2**, 918.

Marek, A. J., Smart, W. D. & Martin, M. C. (2002), Learning visual feature detectors for obstacle avoidance using genetic programming, *in* E. Cantú-Paz, ed., 'Late Breaking Papers at the Genetic and Evolutionary Computation Conference (GECCO-2002)', AAAI, New York, NY, pp. 330–336. URL `http://www.martincmartin.com/papers/LearingVisualFeatureDetectorsForObstAvoidGP_GECCO2002Marek.pdf`.

Martin, D., Fowlkes, C., Tal, D. & Malik, J. (2001), A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, *in* 'Proc. 8th Int'l Conf. Computer Vision', Vol. 2, pp. 416–423.

Matas, J., Chum, O., Urban, M. & Pajdla, T. (2004), 'Robust wide-baseline stereo from maximally stable extremal regions', *Image and Vision Computing* **22**(10), 761 – 767. British Machine Vision Computing 2002.

Maxwell, S. R. (1996), Why might some problems be difficult for genetic programming to find solutions?, *in* J. R. Koza, ed., 'Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996', Stanford Bookstore, Stanford University, CA, USA, pp. 125–128.

McInerney T, T. D. (1996), 'Deformable models in medical image analysis: A survey', *Medical Image Analysis* **1**(2), 91–108.

McKay, B., Willis, M. J. & Barton, G. W. (1995), Using a tree structured genetic algorithm to perform symbolic regression, *in* A. M. S. Zalzala, ed., 'First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA', Vol. 414, IEE, Sheffield, UK, pp. 487–492.

Mikolajczyk, K. & Schmid, C. (2001), Indexing based on scale invariant interest points, *in* 'Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on', Vol. 1, IEEE, pp. 525–531.

Mikolajczyk, K. & Schmid, C. (2002), An affine invariant interest point detector, *in* 'In Proceedings of the 7th European Conference on Computer Vision', pp. 0–7.

Mikolajczyk, K. & Schmid, C. (2004), 'Scale &amp; affine invariant interest point detectors', *International Journal of Computer Vision* **60**, 63–86.

Mikolajczyk, K. & Schmid, C. (2005), 'A performance evaluation of local descriptors', *IEEE transactions on pattern analysis and machine intelligence* **27**(10), 1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. & Gool, L. V. (2005), 'A comparison of affine region detectors', *Int. J. Comput. Vision* **65**, 43–72.

Miller, A. (2002), *Subset selection in regression*, Chapman and Hall/CRC.

Miller, J. F. (1999), An empirical study of the efficiency of learning boolean functions using a cartesian genetic programming approach, *in* W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela & R. E. Smith, eds, 'Proceedings of the Genetic and Evolutionary Computation Conference', Vol. 2, Morgan Kaufmann, Orlando, Florida, USA, pp. 1135–1142. GECCO-99 A joint meeting of the eighth international conference on genetic algorithms (ICGA-99) and the fourth annual genetic programming conference (GP-99).

Milne, M., Raghavendra, P., Leibbrandt, R. & Powers, D. M. W. (2018), 'Personalisation and automation in a virtual conversation skills tutor for children with autism', *Journal on Multimodal User Interfaces* **12**(3), 257–269.

Mirjalili, S., Mirjalili, S. M. & Lewis, A. (2014), 'Grey wolf optimizer', *Advances in engineering software* **69**, 46–61.

Mitchell, M. (1996), *An introduction to genetic algorithms*, MIT Press, Cambridge, MA, USA.

Monsieurs, P. & Flerackers, E. (2003), *Reducing Population Size while Maintaining Diversity*, Vol. 2610 of *Lecture Notes In Computer Science*, Springer-Verlag.

Montesinos, P., Gouet, V., Cedex, F.-N. & Deriche, R. (1998), 'Differential invariants for color images'.

Moravec, H. P. (1977), 'Techniques towards automatic visual obstacle avoidance'.

Moreels, P. & Perona, P. (2005), 'Evaluation of features detectors and descriptors based on 3d objects', *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* **1**, 800–807 Vol. 1.

Moreno, P., Bernardino, A. & Santos-Victor, J. (2006), Model based selection and classification of local features for recognition using gabor filters, *in* A. Campilho & M. Kamel, eds, 'Image Analysis and Recognition', Vol. 4142 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 181–192.

Moreno, R., Puig, D., Julià, C. & Garcia, M. A. (2009), A new methodology for evaluation of edge detectors, *in* 'Image Processing (ICIP), 2009 16th IEEE International Conference on', IEEE, pp. 2157–2160.

Mozos, ., Gil, A., Ballesta, M. & Reinoso, O. (2007), Interest point detectors for visual slam, *in* D. Borrajo, L. Castillo & J. Corchado, eds, 'Current Topics in Artificial Intelligence', Vol. 4788 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 170–179.

Neisser, U. (1964), *Visual search*, Vol. 210, Scientic American.

Neshatian, K., Zhang, M. & Andreae, P. (2012), 'A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming', *IEEE Transactions on Evolutionary Computation* **16**(5), 645–661.

Nikolaev, N. & Iba, H. (2006), *Adaptive Learning of Polynomial Networks Genetic Programming, Backpropagation and Bayesian Methods*, number 4 *in* 'Genetic and Evolutionary Computation', Springer. June.

Nikolic, J., Rehder, J., Burri, M., Gohl, P., Leutenegger, S., Furgale, P. T. & Siegwart, R. (2014), A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam, *in* 'Robotics and Automation (ICRA), 2014 IEEE International Conference on', IEEE, pp. 431–437.

Ojala, T., Pietikainen, M. & Harwood, D. (1994), Performance evaluation of texture measures with classification based on kullback discrimination of distributions, *in* 'Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on', Vol. 1, IEEE, pp. 582–585.

Olague, G. (2016), *Evolutionary computer vision: the first footprints*, Springer.

Olague, G. & Mohr, R. (2002), 'Optimal camera placement for accurate reconstruction', *Pattern Recognition* **35**(4), 927–944.

Olague, G. & Puente, C. (2006), The honeybee search algorithm for three-dimensional reconstruction, *in* 'Workshops on Applications of Evolutionary Computation', Springer, pp. 427–437.

Omran, M. G., Engelbrecht, A. P. & Salman, A. (2004), Image classification using particle swarm optimization, *in* 'Recent Advances in Simulated Evolution and Learning', World Scientific, pp. 347–365.

Owechko, Y. & Medasani, S. (2005), Cognitive swarms for rapid detection of objects and associations in visual imagery, *in* 'Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE', IEEE, pp. 420–423.

Pang, Y., Li, W., Yuan, Y. & Pan, J. (2012), 'Fully affine invariant surf for image matching', *Neurocomputing* **85**, 6–10.

Papari, G. & Petkov, N. (2011), 'Edge and line oriented contour detection: State of the art', *Image and Vision Computing* **29**(2-3), 79–103.

Peirce, C. S. (1884), 'The numerical measure of the success of predictions', *Science* **4**(93), 453–454.

Perdoch, M., Matas, J. & Obdrzalek, S. (2007), Stable affine frames on isophotes, *in* 'Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on Computer Vision', pp. 1–8.

Pereira, A., Vega, J., Moreno, R., Dormido-Canto, S., Rattá, G. A., Pavón, F. & Contributors, J. E. (2015), 'Feature selection for disruption prediction from scratch in jet by using genetic algorithms and probabilistic predictors', *Fusion Engineering and Design* **96**, 907–911.

Perez, C. B. & Olague, G. (2008), Learning invariant region descriptor operators with genetic programming and the f-measure, *in* 'Pattern Recognition, 2008. ICPR 2008. 19th International Conference on', IEEE, pp. 1–4.

Perez, C. B. & Olague, G. (2009*a*), Evolutionary learning of local descriptor operators for object recognition, *in* 'Proceedings of the 11th Annual conference on Genetic and evolutionary computation', ACM, pp. 1051–1058.

Perez, C. B. & Olague, G. (2009*b*), Evolving local descriptor operators through genetic programming, *in* 'Workshops on Applications of Evolutionary Computation', Springer, pp. 414–419.

Perez, C. B. & Olague, G. (2013), 'Genetic programming as strategy for learning image descriptor operators', *Intelligent Data Analysis* **17**(4), 561–583.

Perruchet, P. & Peereman, R. (2004), 'The exploitation of distributional information in syllable processing', *Journal of Neurolinguistics* **17**(2-3), 97–119.

Petrou, M. & Kittler, J. (1991), 'Optimal edge detectors for ramp edges', *IEEE Transactions on Pattern Analysis & Machine Intelligence* (5), 483–491.

Plotkin, H. C. (1993), *Darwin machines and the nature of knowledge*, Cambridge, MA: Harvard University Press.

Poli, R. (1996*a*), Discovery of symbolic, neuro-symbolic and neural networks with parallel distributed genetic programming, Technical Report CSRP-96-14, University of Birmingham, School of Computer Science. Presented at 3rd International Conference on Artificial Neural Networks and Genetic Algorithms, ICANNGA'97.

Poli, R. (1996*b*), Genetic programming for feature detection and image segmentation, *in* 'Selected Papers from AISB Workshop on Evolutionary Computing', Springer-Verlag, London, UK, pp. 110–125.

Poli, R. (1996*c*), Genetic programming for image analysis, *in* J. R. Koza, D. E. Goldberg, D. B. Fogel & R. L. Riolo, eds, 'Genetic Programming 1996: Proceedings of the First Annual Conference', MIT Press, Stanford University, CA, USA, pp. 363–368. GP-96.

Poli, R. (1999), Parallel distributed genetic programming, *in* D. Corne, M. Dorigo & F. Glover, eds, 'New Ideas in Optimization', Advanced Topics in Computer Science, McGraw-Hill, Maidenhead, Berkshire, England, chapter 27, pp. 403–431.

Poli, R., Kennedy, J. & Blackwell, T. (2007), 'Particle swarm optimization', *Swarm intelligence* **1**(1), 33–57.

Poli, R. & Langdon, W. B. (1998*a*), On the search properties of different crossover operators in genetic programming, *in* J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba & R. Riolo, eds, 'Genetic Programming 1998: Proceedings of the Third Annual Conference', Morgan Kaufmann, University of Wisconsin, Madison, Wisconsin, USA, pp. 293–301.

Poli, R. & Langdon, W. B. (1998*b*), 'Schema theory for genetic programming with one-point crossover and point mutation', *Evolutionary Computation* **6**(3), 231–252.

Poli, R., Langdon, W. B. & Dignum, S. (2007), On the limiting distribution of program sizes in tree-based genetic programming, *in* M. Ebner, M. O'Neill, A. Ekárt, L. Vanneschi & A. I. Esparcia-Alcázar, eds, 'Proceedings of the 10th European Conference on Genetic Programming', Vol. 4445 of *Lecture Notes in Computer Science*, Springer, Valencia, Spain, pp. 193–204.

Poli, R., Langdon, W. B. & McPhee, N. F. (2008), *A field guide to genetic programming*, Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`. With contributions by J. R. Koza.

Posada, J., Toro, C., Barandiaran, I., Oyarzun, D., Stricker, D., De Amicis, R., Pinto, E. B., Eisert, P., Döllner, J. & Vallarino, I. (2015), 'Visual computing as a key enabling technology for industrie 4.0 and industrial internet', *IEEE computer graphics and applications* **35**(2), 26–40.

Potkonjak, V., Gardner, M., Callaghan, V., Mattila, P., Guetl, C., Petrović, V. M. & Jovanović, K. (2016), 'Virtual laboratories for education in science, technology, and engineering: A review', *Computers & Education* **95**, 309–327.

Powers, D. M. (2011), 'Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation'.

Powers, D. M. (2012*a*), The problem with kappa, *in* 'Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics', Association for Computational Linguistics, pp. 345–355.

Powers, D. M. (2013), 'Adabook & multibook'.

Powers, D. M. (2015*a*), 'What the f-measure doesn't measure: Features, flaws, fallacies and fixes', *arXiv preprint arXiv:1503.06410* .

Powers, D. M. W. (2012*b*), Roc-concert: Roc-based measurement of consistency and certainty, *in* '2012 Spring Congress on Engineering and Technology', pp. 1–4.

Powers, D. M. W. (2015*b*), 'Evaluation evaluation a monte carlo study', *CoRR* **abs/1504.00854**.
URL: *http://arxiv.org/abs/1504.00854*

Powers, D. M. W. (2015*c*), 'Evaluation evaluation a monte carlo study', *CoRR* **abs/1504.00854**.
URL: *http://arxiv.org/abs/1504.00854*

Powers, D. M. W. (2015*d*), 'Visualization of tradeoff in evaluation: from precision-recall & PN to lift, ROC & BIRD', *CoRR* **abs/1505.00401**.
URL: *http://arxiv.org/abs/1505.00401*

Prewitt, J. M. (1970), 'Object enhancement and extraction', *Picture processing and Psychopictorics* **10**(1), 15–19.

Price, K., Storn, R. M. & Lampinen, J. A. (2006), *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media.

Quinlan, J. R. (1986), 'Induction of decision trees', *Machine learning* **1**(1), 81–106.

Rahmani, H., Mahmood, A., Huynh, D. Q. & Mian, A. (2014), Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition, *in* 'European conference on computer vision', Springer, pp. 742–757.

Rechenberg, I. (1965), 'Cybernetic solution path of an experimental problem', Ministry of Aviation, Royal Aircraft Establishment (U.K.).

Reyes, M. E. P. & Chen, S.-C. (2017), A 3d virtual environment for storm surge flooding animation, *in* '2017 IEEE third international conference on multimedia big data (BigMM)', IEEE, pp. 244–245.

Rezende, D. J., Eslami, S. A., Mohamed, S., Battaglia, P., Jaderberg, M. & Heess, N. (2016), Unsupervised learning of 3d structure from images, *in* 'Advances in Neural Information Processing Systems', pp. 4996–5004.

Rietzler, M., Plaumann, K., Kränzle, T., Erath, M., Stahl, A. & Rukzio, E. (2017), Vair: Simulating 3d airflows in virtual reality, *in* 'Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems', ACM, pp. 5669–5677.

Rijsbergen, C. J. V. (1979), *Information Retrieval*, 2nd edn, Butterworth-Heinemann, Newton, MA, USA.

Roberts, L. G. (1963), Machine perception of three-dimensional solids, PhD thesis, Massachusetts Institute of Technology.

Rodriguez-Coayahuitl, L., Morales-Reyes, A. & Escalante, H. J. (2018), Structurally layered representation learning: Towards deep learning through genetic programming, *in* 'European Conference on Genetic Programming', Springer, pp. 271–288.

Rohr, K. (1992), Modelling and identification of characteristic intensity variations, *in* 'Image and Vision Computing', Vol. 10, pp. 66–76.

Rosin, P. L. (1999), 'Measuring corner properties', *Comput. Vis. Image Underst.* **73**(2), 291–307.

Rosten, E. & Drummond, T. (2006), Machine learning for high-speed corner detection, *in* 'European Conference on Computer Vision', Vol. 1, pp. 430–443.

Rosten, E., Porter, R. & Drummond, T. (2010), 'Faster and better: A machine learning approach to corner detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(1), 105–119.

Russell, S. J. & Norvig, P. (2016), *Artificial intelligence: a modern approach*, Malaysia; Pearson Education Limited,.

Rusu, R. B., Blodow, N. & Beetz, M. (2009), Fast point feature histograms (fpfh) for 3d registration, *in* 'Robotics and Automation, 2009. ICRA'09. IEEE International Conference on', Citeseer, pp. 3212–3217.

Rusu, R. B., Blodow, N., Marton, Z. C. & Beetz, M. (2008), Aligning point cloud views using persistent feature histograms, *in* 'Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on', IEEE, pp. 3384–3391.

Sasaki, Y. et al. (2007), 'The truth of the f-measure', *Teach Tutor mater* **1**(5), 1–5.

Schaffalitzky, F. & Zisserman, A. (2002), Multi-view matching for unordered image sets, or how do i organize my holiday snaps?, *in* 'European conference on computer vision', Springer, pp. 414–431.

Schiele, B. & Kruppa, H. (2003), 'Using local context to improve face detection'.

Schmid, C., Mohr, R. & Bauckhage, C. (2000), 'Evaluation of interest point detectors', *International Journal of Computer Vision* **37**, 151–172.

Schönauer, M., Sebag, M., Jouve, F., Lamy, B. & Maitournam, H. (1996), Evolutionary identification of macro-mechanical models, *in* P. J. Angeline & K. E. Kinnear, Jr., eds, 'Advances in Genetic Programming 2', MIT Press, Cambridge, MA, USA, pp. 467–488.

Sebastiani, F. (2002), 'Machine learning in automated text categorization', *ACM computing surveys (CSUR)* **34**(1), 1–47.

Sebe, N., Cohen, I., Garg, A. & Huang, T. S. (2005), *Machine Learning in Computer Vision (Computational Imaging and Vision)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Shannon, C. E. & Weaver, W. (1949), 'The mathematical theory of communication (champaign, il', *Urbana: University of Illinois Press* .

Shao, L., Liu, L. & Li, X. (2014), 'Feature learning for image classification via multiobjective genetic programming', *IEEE Transactions on Neural Networks and Learning Systems* **25**(7), 1359–1371.

Singh, T., Kharma, N., Daoud, M. & Ward, R. (2009), Genetic programming based image segmentation with applications to biomedical object detection, *in* 'Proceedings of the 11th Annual conference on Genetic and evolutionary computation', ACM, pp. 1123–1130.

Sipiran, I. & Bustos, B. (2010), A robust 3d interest points detector based on harris operator, *in* 'Eurographics 2010 Workshop on 3D Object Retrieval', The Eurographics Association, pp. 7–14.

Sipiran, I. & Bustos, B. (2011), 'Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes', *The Visual Computer* **27**(11), 963.

Smart, W. & Zhang, M. (2003), Classification strategies for image classification in genetic programming, *in* 'Proceeding of image and vision computing conference', Palmerston North, New Zealand, pp. 402–407.

Smith, S. M. & Brady, J. M. (1997), 'Susana new approach to low level image processing', *International journal of computer vision* **23**(1), 45–78.

Sobel, I. & Feldman, G. (1968), 'A 3x3 isotropic gradient operator for image processing', *a talk at the Stanford Artificial Project in* pp. 271–272.

Song, A. & Ciesielski, V. (2008), 'Texture segmentation by genetic programming', *Evolutionary Computation* **16**(4), 461–481.

Song, A., Ciesielski, V. & Williams, H. E. (2002), Texture classifiers generated by genetic programming, *in* 'Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on', Vol. 1, IEEE, pp. 243–248.

Song, A., Loveard, T. & Ciesielski, V. (2001), Towards genetic programming for texture classification, *in* 'Australian joint conference on artificial intelligence', Springer, pp. 461–472.

Sportillo, D., Paljic, A., Boukhris, M., Fuchs, P., Ojeda, L. & Roussarie, V. (2017), An immersive virtual reality system for semi-autonomous driving simulation: a comparison between realistic and 6-dof controller-based interaction, *in* 'Proceedings of the 9th International Conference on Computer and Automation Engineering', ACM, pp. 6–10.

Stanford 3D Scanning Repository (n.d.).
URL: *http://graphics.stanford.edu/data/3Dscanrep/*

Steder, B., Rusu, R. B., Konolige, K. & Burgard, W. (2011), Point feature extraction on 3d range scans taking into account object boundaries, *in* 'Robotics and automation (icra), 2011 ieee international conference on', IEEE, pp. 2601–2608.

Storn, R. & Price, K. (1997), 'Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces', *Journal of global optimization* **11**(4), 341–359.

Stottinger, J., Hanbury, A., Sebe, N. & Gevers, T. (2012), 'Sparse color interest points for image retrieval and object categorization', *IEEE Transactions on Image Processing* **21**(5), 2681–2692.

Tabbone, S. & Lorraine, C. (1993), Corner detection using laplacian of gaussian operator, *in* 'Proceedings of the Scandanavian Conference on Image Analysis', Vol. 2, pp. 1055–1055.

Tackett, W. A. (1993), Genetic programming for feature discovery and image discrimination, *in* 'Proceedings of the Fifth International Conference on Genetic Algorithms', Morgan Kaufmann, pp. 303–309.

Tan, T. (1995), 'Texture edge detection by modelling visual cortical channels', *Pattern Recognition* **28**(9), 1283–1298.

Teller, A. (1996), Evolving programmers: The co-evolution of intelligent recombination operators, *in* P. J. Angeline & K. E. Kinnear, Jr., eds, 'Advances in Genetic Programming 2', MIT Press, Cambridge, MA, USA, chapter 3, pp. 45–68. PADO + SMART recombination html version available from http://www.cs.cmu.edu/ astro/.

Thomas Bäck, Günter Rudolph, H.-P. S. (1993), 'Theory of genetic algorithms', *Current Trends in Theoretical Computer Science* **1**, 546–578.

Tombari, F., Salti, S. & Di Stefano, L. (2010*a*), Unique shape context for 3d data description, *in* 'Proceedings of the ACM workshop on 3D object retrieval', ACM, pp. 57–62.

Tombari, F., Salti, S. & Di Stefano, L. (2010*b*), Unique signatures of histograms for local surface description, *in* 'European conference on computer vision', Springer, pp. 356–369.

Tran, B., Xue, B. & Zhang, M. (2016), 'Genetic programming for feature construction and selection in classification on high-dimensional data', *Memetic Computing* **8**(1), 3–15.

Trujillo, L., Legrand, P., Olague, G. & Pérez, C. (2010), Optimization of the hölder image descriptor using a genetic algorithm, *in* 'Proceedings of the 12th annual conference on Genetic and evolutionary computation', ACM, pp. 1147–1154.

Trujillo, L. & Olague, G. (2006), Synthesis of interest point detectors through genetic programming, *in* 'Proceedings of the 8th annual conference on Genetic and evolutionary computation', ACM, pp. 887–894.

Trujillo, L. & Olague, G. (2008), 'Automated design of image operators that detect interest points', *Massachusetts Institute of Technology* **16**(4), 483–507.

Tuytelaars, T. & Mikolajczyk, K. (2008), 'Local invariant feature detectors: a survey', *Found. Trends. Comput. Graph. Vis.* **3**(3), 177–280.

Ullmann, J. R. & Kidd, P. (1969), 'Recognition experiments with typed numerals from envelopes in the mail', *Pattern Recognition* **1**(4), 273–289.

Viola, P. A. & Jones, M. J. (2004), 'Robust real-time face detection', *International Journal of Computer Vision* **57**, 137–154.

Viola, P. & Jones, M. (2001*a*), 'Rapid object detection using a boosted cascade of simple features', *In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp. 905–910.

Viola, P. & Jones, M. (2001*b*), 'Robust real-time object detection', *International Journal of Computer Vision* .

Wiegand, R. P. (2004), An analysis of cooperative coevolutionary algorithms, PhD thesis, George Mason University, Fairfax, VA, USA. Director-Jong, Kenneth A.

Williams-Bell, F. M., Kapralos, B., Hogue, A., Murphy, B. & Weckman, E. (2015), 'Using serious games and virtual simulation for training in the fire service: a review', *Fire Technology* **51**(3), 553–584.

Wilson, S. W. (1999), Get real! xcs with continuous-valued inputs, *in* 'International Workshop on Learning Classifier Systems', Springer, pp. 209–219.

Winder, S. A. & Brown, M. (2007), Learning local image descriptors, *in* 'Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on', IEEE, pp. 1–8.

Winkeler, J. F. & Manjunath, B. (1997), 'Genetic programming for object detection', *Genetic Programming* pp. 330–335.

Witkin, A. P. (1987), Scale-space filtering, *in* 'Readings in Computer Vision', Elsevier, pp. 329–332.

Woodward, P. M. (1953), *Probability and information theory with applications to radar*, London: Pergamon Press.

Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T. S. & Yan, S. (2010), 'Sparse representation for computer vision and pattern recognition', *Proceedings of the IEEE* **98**(6), 1031–1044.

Wu, J., Xue, T., Lim, J. J., Tian, Y., Tenenbaum, J. B., Torralba, A. & Freeman, W. T. (2016), Single image 3d interpreter network, *in* 'European Conference on Computer Vision', Springer, pp. 365–382.

Xue, B., Zhang, M., Browne, W. N. & Yao, X. (2016), 'A survey on evolutionary computation approaches to feature selection', *IEEE Transactions on Evolutionary Computation* **20**(4), 606–626.

Xue, P., Pal, J. S., Ye, X., Lenters, J. D., Huang, C. & Chu, P. Y. (2017), 'Improving the simulation of large lakes in regional climate modeling: Two-way lake–atmosphere coupling with a 3d hydrodynamic model of the great lakes', *Journal of Climate* **30**(5), 1605–1627.

Yang, X.-S. (2013), Bat algorithm and cuckoo search: a tutorial, *in* 'Artificial Intelligence, Evolutionary Computing and Metaheuristics', Springer, pp. 421–434.

Yi, K. M., Trulls, E., Lepetit, V. & Fua, P. (2016), Lift: Learned invariant feature transform, *in* 'European Conference on Computer Vision', Springer, pp. 467–483.

Youden, W. J. (1950), 'Index for rating diagnostic tests', *Cancer* **3**(1), 32–35.

Yussof, W. N. J. H. W. & Hitam, M. S. (2014), 'Invariant gabor-based interest points detector under geometric transformation', *Digital Signal Processing* **25**, 190–197.

Zhang, M., Cagnoni, S. & Olague, G. (2009), Evolutionary computer vision, *in* 'Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers', ACM, pp. 3355–3380.

Zhang, W.-C. & Shui, P.-L. (2015), 'Contour-based corner detection via angle difference of principal directions of anisotropic gaussian directional derivatives', *Pattern Recognition* **48**(9), 2785–2797.

Zhang, Y. & Rockett, P. I. (2005), 'Evolving optimal feature extraction using multi-objective genetic programming: A methodology and preliminary study on edge detection'.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N. & Zhang, Q. (2011), 'Multiobjective evolutionary algorithms: A survey of the state of the art', *Swarm and Evolutionary Computation* **1**(1), 32–49.

Zhu, Y., Chen, W. & Guo, G. (2014), 'Evaluating spatiotemporal interest point features for depth-based action recognition', *Image and Vision Computing* **32**(8), 453–464.