Linköping studies in science and technology. Dissertations.
No. 1811

Embodied Visual Object Recognition

Marcus Wallenberg

LINKÖPING UNIVERSITY

Cover illustration: *At night I stand alone in a darkened room, surrounded by unknown objects*. Marcus Wallenberg, 2016.

Linköping studies in science and technology. Dissertations.
No. 1811

Embodied Visual Object Recognition

Marcus Wallenberg

marcus.wallenberg@liu.se
www.cvl.isy.liu.se
Computer Vision Laboratory
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 978-91-7685-626-0 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2017

For Lisa

Abstract

Object recognition is a skill we as humans often take for granted. Due to our formidable object learning, recognition and generalisation skills, it is sometimes hard to see the multitude of obstacles that need to be overcome in order to replicate this skill in an artificial system. Object recognition is also one of the classical areas of computer vision, and many ways of approaching the problem have been proposed.

Recently, visually capable robots and autonomous vehicles have increased the focus on embodied recognition systems and active visual search. These applications demand that systems can learn and adapt to their surroundings and arrive at decisions in a reasonable amount of time. Ideally, this should be done while maintaining high object recognition performance. This is especially challenging due to the high dimensionality of image data and in cases where end-to-end learning from pixels to output is needed. Therefore, mechanisms designed to make inputs tractable are often necessary for less computationally capable embodied systems.

Active visual search also means that mechanisms for attention and gaze control are integral to the object recognition procedure. Therefore, the way in which attention mechanisms should be introduced into feature extraction and estimation algorithms must be carefully considered when constructing a recognition system.

This thesis describes work done on the components necessary for creating an embodied recognition system, specifically in the areas of decision uncertainty estimation, object segmentation from multiple cues, adaptation of stereo vision to a specific platform and setting, problem-specific feature selection, efficient estimator training and attentional modulation in convolutional neural networks. Contributions include the evaluation of methods and measures for predicting the potential uncertainty reduction that can be obtained from additional views of an object, allowing for adaptive target observations. Also, in order to separate a specific object from other parts of a scene, it is often necessary to combine multiple cues such as colour and depth in order to obtain satisfactory results. Therefore, a method for combining these using channel coding has been evaluated. In order to make use of three-dimensional spatial structure in recognition, a novel stereo vision algorithm extension along with a framework for automatic stereo tuning have also been investigated. Feature selection and efficient discriminant sampling for decision tree-based estimators have also been implemented. Finally, attentional multi-layer modulation of convolutional neural networks for recognition in cluttered scenes has been investigated. Several of these components have been tested and evaluated on a purpose-built embodied recognition platform known as Eddie the Embodied.

Populärvetenskaplig sammanfattning

Förmågan att tolka och lära av synintryck är både viktig och självklar för oss människor. Detta gör att vi sällan ägnar en tanke åt den enorma komplexitet och de många utmaningar vi med lätthet hanterar varje gång vi upptäcker eller känner igen en plats, ett föremål eller en person. Vid utformning av artificiella igenkänningssystem utgör dock dessa utmaningar stora problem, som ännu delvis saknar lösning.

Inom bildanalys och datorseende har dessa mekanismer studerats länge, och många förslag till lösningar har presenterats. I takt med att nya tekniker för dataanalys och maskininlärning utvecklats, har dessa också fått en alltmer betydande roll för igenkänningssystem. När seende robotar nu förbereds för sin plats i såväl industriella som vardagliga sammanhang är praktiska lösningar på igenkänningsproblemet viktigare än någonsin.

Förkroppsligade system ställs inför ett igenkänningsscenario som skiljer sig från mycket av den tidigare forskningen inom området. De måste ofta fatta beslut baserade på begränsade mängder data, med begränsad beräkningskraft och inom begränsad tid. Förkroppsligade system har dock möjligheten att utforska och interagera med sin omgivning för att förbättra underlaget för sådana beslut. De har även möjligheten att anpassa sig till specifika omständigheter. Denna kombination av bredd (anpassningsförmåga i olika situationer) och djup (möjligheten att vid varje situation anpassa sig till omständigheterna) skiljer förkroppsligade igenkänningssystem från statiska databasbaserade metoder. Moderna inlärningsmetoder för igenkänningssystem är kapabla att skapa mycket komplexa kopplingar direkt från bilddata till tolkning. Detta innebär dock stora minnes- och beräkningskostnader på grund av den höga dimensionalitet bilddata nödvändigtvis medför, och praktiska sätt att hantera detta måste beaktas vid systemets konstruktion. Visuell avsökning och tolkning av scener innebär också att mekanismer för uppmärksamhet och blickstyrning spelar en viktig roll, och påverkar systemets alla nivåer.

Denna avhandling tar upp arbete relaterat till komponenter i ett förkroppsligat igenkänningssystem, och vissa specifika aspekter av deras individuella funktioner. Särskilt behandlas osäkerhetsskattning vid beslutsfattande över tid, anpassning av stereoskopiskt seende till specifika situationer, problemspecifika val av särdrag, effektiva inlärningsmetoder och visuell uppmärksamhet och blickstyrning i artificiella neurala nätverk.

Vetenskapliga bidrag och resultat är främst: Utvärdering av metoder för att skatta minskningen i osäkerhet som nya observationer av ett föremål kan ge, vilket i sin tur kan ligga till grund för beslutsfattande; segmenteringsmetoder för att skilja ett föremål från sin omgivning baserad på kanalrepresentationer av färg och avstånd; en förbättrad metod för stereomatchning i ett tvåkamerasystem, och en metod för att anpassa denna till specifika situationer; val av särdrag och diskriminanter för effektiv träning av beslutsträd; uppmärksamhetsmodulering för djupa faltningsnätverk för igenkänning i komplexa scener. Flertalet av dessa metoder har även utvärderats på en särskilt framtagen förkroppsligad igenkänningsplattform kallad “Eddie the Embodied".

Acknowledgments

At a recent PopSci event, I was approached by one of the other exhibitors and told that what had tipped the scales and made her go into engineering was a demonstration of Eddie at a similar event several years prior. Likewise, my own endeavours into computer vision research have been inspired and influenced by a great many people in a great many ways. From professional and scientific advice to inspiration for guitar riffs and, at one time, the donation of an organ[1], these have all played their parts in bringing this work and its author to their current state. Although too numerous to mention individually, I would like to extend my special thanks to:

- My supervisors Per-Erik Forssén and Michael Felsberg, for giving me the trust and opportunity to carry out this work, and for their continued support throughout.
- Everyone at CVL (past and present) whose knowledge and ideas were vital in providing insight, support and inspiration.
- My wife Lisa, for not letting me give up despite professional and personal setbacks, and my own serious doubts about the outcome of this work.
- My parents, siblings and relatives for encouraging me pursue my research ambitions.
- All the friends I have made since coming to Linköping. To Oskar, Ulf, Andreas, Henrik, Jonas, Tore, Jakob, Rikard, Peter, Markus, Mattias and everyone else that I'm forgetting to mention here.

This work was supported by the Swedish Research Council through a grant for the project Embodied Visual Object Recognition, by VINNOVA through the FaceTrack project and by Linköping University.

Linköping, November 2016
Marcus Wallenberg

[1] Of the musical variety.

Contents

I Background

1 Introduction **3**
- 1.1 Motivation . . . 3
  - 1.1.1 This thesis . . . 4
  - 1.1.2 Outline Part I: Background Theory . . . 4
  - 1.1.3 Outline Part II: Included Publications . . . 6

2 What is embodied visual object recognition? **13**
- 2.1 A recognition system in the real world . . . 14
- 2.2 How do embodied systems learn? . . . 14
- 2.3 Building an artificial embodied recognition system . . . 15

3 Eyes and vision **17**
- 3.1 The camera-type eye . . . 17
- 3.2 Peripheral and foveal vision . . . 18
- 3.3 Saccades and fixations . . . 19
- 3.4 Spatial information, vergence and parallax . . . 20

4 Single and multiple view geometry **21**
- 4.1 Single-view geometry and the pinhole camera . . . 22
  - 4.1.1 Thin-lens cameras and their pinhole approximations . . . 22
  - 4.1.2 Digital cameras and the pixel grid . . . 23
- 4.2 The effects of lens distortion . . . 24
  - 4.2.1 Common types of lens distortion . . . 24
  - 4.2.2 Lens distortion in the single-camera case . . . 25
- 4.3 Multiple view geometry . . . 27

5 Stereo Vision **29**
- 5.1 Epipolar geometry . . . 29
  - 5.1.1 Epipolar geometry and lens distortion . . . 31
- 5.2 Stereo vision algorithms . . . 32

5.2.1 Global versus local methods 32
5.2.2 Correspondence propagation 33
5.2.3 Coarse-to-Fine Best-First Propagation 35
5.3 Structured light systems 35

6 Visual attention **37**
6.1 What to look at 37
6.1.1 The concept of visual saliency 37
6.2 Algorithms for saliency detection 38
6.3 Dynamic visual attention and inhibition of return 39

7 Segmentation **41**
7.1 Where to draw the line - the concept of objects 41
7.2 Segmentation and image representation 42
7.3 What is good segmentation? 43
7.3.1 Performance measures 43

8 Description, Learning and Representation **45**
8.1 What is image content anyway? 45
8.1.1 The descriptiveness-invariance trade-off 46
8.1.2 Dense versus sparse representations 46
8.1.3 Ordered and unordered representations 47
8.2 Commonly used descriptors 47
8.3 Learning and inference 48
8.3.1 Bag-of-Words methods 49
8.3.2 Single-classifiers and ensemble methods 49
8.3.3 Neural networks and deep learning 50

9 Bag-of-words methods **51**
9.1 The Bag-of-Words method in computer vision 51
9.1.1 Vocabulary generation 52
9.1.2 Learning 52
9.2 Confidence and hesitation 53
9.2.1 Confidence measures 53
9.2.2 Confidence gain and hesitation 54

10 Decision forests **55**
10.1 Decision trees and decision forests 55
10.1.1 Decision trees 55
10.1.2 Decision forests 56
10.2 Feature selection and learning 58
10.2.1 Bagging 58
10.3 CCA-based feature selection 58
10.3.1 Discriminant selection 59

11 Neural networks and deep learning **61**
11.1 Artificial neural networks 61
11.1.1 Convolutional neural networks 62
11.1.2 Basic operations and structure 63
11.1.3 The rise of deep learning 64
11.1.4 Pre-trained deep networks for feature extraction 66
11.2 Attention models in deep networks 67
11.2.1 Bottom-up attention for pre-trained deep networks 68

12 Eddie: an EVOR platform **71**
12.1 Hardware description 71
12.2 Software control structure during the EVOR project 74
12.2.1 Attention and visuomotor control 74
12.2.2 Learning and recognition 76
12.3 Wide-angle stereo calibration and tuning 76
12.3.1 Point-to-point mappings 76
12.3.2 Error variance propagation and weighting 77
12.3.3 Calibration procedure 79
12.4 Later experiments 79

13 Concluding Remarks **81**

Bibliography **83**

II Publications

A A Research Platform for Embodied Visual Object Recognition **93**

B Embodied Object Recognition using Adaptive Target Observations **99**

C Channel Coding for Joint Colour and Depth Segmentation **111**

D Teaching Stereo Perception to YOUR Robot **123**

E Improving Random Forests by correlation-enhancing projections and sample-based sparse discriminant selection **137**

F Attention Masking for Pre-trained Deep Networks **145**

Part I

Background

1

Introduction

1.1 Motivation

Object learning and recognition is such an integral part of our daily life that we seldom ponder the vast complexity of the recognition tasks we perform, seemingly with no effort at all. Although great advances have been made in recent years, most artificial systems can at best utilise purpose-built software and hardware to perform highly specialised recognition tasks. Most artificial systems cannot rival their biological counterparts in speed, robustness or ability to generalise simultaneously, and those that excel in any of these aspects are usually extremely computationally expensive. This however, does not mean that human visual cognition (or that of any other organism for that matter) is a "perfect" solution. As we all know, looks (or in this case, rather *vision*) can be deceiving, and many examples of optical illusions and hallucinated visions indicate that much of what we "see" is based on preconceived notions and assumptions rather than actual visual input.

Object learning and recognition are also necessary skills for artificial cognitive systems if they are to be of use to us. Almost any task in our everyday lives requires knowledge of objects and their properties. A task such as "go fetch my slippers from the cupboard in the hallway" requires knowledge of not only the slippers themselves, but the topological relationships between the slippers, the cupboard and the hallway. The meaning of the "fetching" action is also necessary, and the ability to determine whose slippers are to be retrieved also requires that the system can tell people apart. Thus, the complexity of the object recognition task and the great demands it places on other cognitive processes requires a holistic approach that, explicitly or implicitly, takes all of these factors into account.

1.1.1 This thesis

The aim of this thesis is to summarise and put into perspective work done on the components of robot vision for embodied object recognition (in the Embodied Visual Object Recognition (EVOR) project, 2009-2013) and subsequent investigations into related applications and techniques such as attribute estimation using decision forests (the FaceTrack project, 2013-2015) and the incorporation of attentional masking in pre-trained deep neural networks (2015-2016).

The aim of the original EVOR project was to investigate aspects of embodied object recognition on a robotic platform. A recognition system was to be implemented and evaluated, and the basic building blocks studied both separately and in a combined online setting. Questions relating to camera geometry and calibration, stereo vision, object learning and segmentation were the subject of the author's contributions to the project. At the close of the project, a basic system had been devised. Focus was then shifted to improving the learning capabilities of the system by investigating more sophisticated machine learning techniques. The author's contributions to this end consisted of devising feature selection and learning procedures for use with decision forests and pre-trained deep neural networks.

1.1.2 Outline Part I: Background Theory

The background theory and introduction portion of this thesis contains the following chapters:

- Chapter 2: *What is object recognition?* - in which the topic of embodied object recognition in both naturally occurring and artificial systems is outlined. This also serves as an introduction to the techniques described later, placing them in the context of a recognition setting.
- Chapter 3: *Eyes and vision* - in which basic concepts of eyes and vision are described.
- Chapter 4: *Single and multiple view geometry* - in which the basics of camera models and camera geometry are presented.
- Chapter 5: *Stereo vision* - in which the principles of the two-camera stereo problem are described. Also included are brief descriptions of stereo algorithms and a more in-depth description of correspondence propagation.
- Chapter 6: *Visual attention* - in which functions of visual attention, the concept of visual saliency and mechanisms of visual search are described.
- Chapter 7: *Segmentation* - in which the concept of image segmentation and the importance of representation when fusing multiple measurement modalities is discussed.
- Chapter 8: *Description, learning and representation* - in which the concepts of structure description, invariance and examples of object learning are introduced.

- Chapter 9: *Bag-of-Words methods* - in which the principles of Bag-of-Words matching used in the EVOR project are described.
- Chapter 10: *Decision forests* - in which the principles of decision forests used in the FaceTrack project are described.
- Chapter 11: *Neural networks and deep learning* - in which the principles of artificial neural networks and the concept of attention in this context are described.
- Chapter 12: *Eddie: An EVOR platform* - in which the platform used in the EVOR project is described, and its functionalities and design choices are discussed.
- Chapter 13: *Concluding remarks* - in which conclusions drawn and insights gained during this work are discussed.

1.1.3 Outline Part II: Included Publications

Selected versions of six manuscripts are included in Part II. The full details and abstracts of these papers together with statements of their relevance, as well as contributions made by the author, are summarised below.

Paper A: A Research Platform for Embodied Visual Object Recognition

> Marcus Wallenberg and Per-Erik Forssén. A Research Platform for Embodied Visual Object Recognition. In *Proceedings of SSBA 2010 Symposium on Image Analysis*, pages 137–140, 2010b.

Abstract:
We present in this paper a research platform for development and evaluation of embodied visual object recognition strategies. The platform uses a stereoscopic peripheral-foveal camera system and a fast pan-tilt unit to perform saliency-based visual search. This is combined with a classification framework based on the bag-of-features paradigm with the aim of targeting, classifying and recognising objects. Interaction with the system is done via typed commands and speech synthesis. We also report the current classification performance of the system.
Relevance and contribution:
In order to study embodied visual object recognition, the basic building blocks of such a system must be implemented. This system combines the elements necessary for wide-angle stereo vision and attention, as well as rudimentary gaze control. The system also has mechanisms for communicating with the user through voice synthesis and typed commands. The system implements a well-known method of object recognition based on sparse keypoints. Although the specific method used for matching is no longer as common, the basic principles of this system should be useful to anyone wanting to study embodied visual object recognition.

The author has contributed to the hardware and software implementation of the object recognition platform, and designed and implemented procedures for data capture and evaluation of the system as well as contributed to the writing of the paper.

Paper B: Embodied Object Recognition using Adaptive Target Observations

Marcus Wallenberg and Per-Erik Forssén. Embodied Object Recognition using Adaptive Target Observations. *Cognitive Computation*, 2 (4):316–325, 2010a.

Abstract:
In this paper, we study object recognition in the embodied setting. More specifically, we study the problem of whether the recognition system will benefit from acquiring another observation of the object under study, or whether it is time to give up, and report the observed object as unknown. We describe the hardware and software of a system that implements recognition and object permanence as two nested perception-action cycles. We have collected three data sets of observation sequences that allow us to perform controlled evaluation of the system behaviour. Our recognition system uses a KNN classifier with bag-of-features prototypes. For this classifier, we have designed and compared three different uncertainty measures for target observation. These measures allow the system to (a) decide whether to continue to observe an object or to move on, and to (b) decide whether the observed object is previously seen or novel. The system is able to successfully reject all novel objects as "unknown", while still recognising most of the previously seen objects.
Relevance and contribution:
The concept of confidence is often treated as a one-time measurement, providing a single confidence value for a putative classification. This is indeed a valid approach when classifying single images. However, in an embodied real-time system, the change in confidence resulting from the aggregation of additional views provides information about the predicted value of a new observation. This should be of importance when studying planning behaviours and confidence in an embodied real-time setting. Although the exact method used for matching will determine the exact nature of the confidence measure, the principles should be applicable to a variety of learning systems.

In addition to contributing to the writing of the paper, the author has contributed to the design of the confidence and hesitation measures and their incorporation into the object recognition system. The author has also contributed to the evaluation methodology and carried out data capture and experimental evaluation of the system.

Paper C: Channel Coding for Joint Colour and Depth Segmentation

Marcus Wallenberg, Michael Felsberg, Per-Erik Forssén, and Babette Dellen. Channel Coding for Joint Colour and Depth Segmentation. In *Proceedings of the DAGM Symposium on Pattern Recognition*, 2011a

Abstract:
Segmentation is an important preprocessing step in many applications. Compared to colour segmentation, fusion of colour and depth greatly improves the segmentation result. Such a fusion is easy to do by stacking measurements in different value dimensions, but there are better ways. In this paper we perform fusion using the channel representation, and demonstrate how a state-of-the-art segmentation algorithm can be modified to use channel values as inputs. We evaluate segmentation results on data collected using the Microsoft Kinect peripheral for Xbox 360, using the superparamagnetic clustering algorithm. Our experiments show that depth gradients are more useful than depth values for segmentation, and that channel coding both colour and depth gradients makes tuned parameter settings generalise better to novel images.

Relevance and contribution:
In order to enable a recognition system to deal with objects, it must be able to distinguish between objects and their surroundings. Thus, a robust method for segmenting potentially very similar objects from each other and their background is needed. This method can make use of intensity, colour and spatial information and is presented along with performance measures and an evaluation framework for adapting it to an appropriate setting.

The author has contributed to the design of the performance measures used and implemented procedures for data capture, experimental evaluation and optimisation of the final algorithm. The author has also contributed to the writing of the paper.

Paper D: Teaching Stereo Perception to YOUR Robot

Marcus Wallenberg and Per-Erik Forssén. Teaching Stereo Perception to YOUR Robot. In *Proceedings of the British Machine Vision Conference*, 2012

Abstract:
This paper describes a method for generation of dense stereo ground-truth using a consumer depth sensor such as the Microsoft Kinect. Such ground-truth allows adaptation of stereo algorithms to a specific setting. The method uses a novel residual weighting based on error propagation from image plane measurements to 3D. We use this groundtruth in wide-angle stereo learning by automatically tuning a novel extension of the best-first propagation (BFP) dense correspondence algorithm. We extend BFP by adding a coarse-to-fine scheme, and a structure measure that limits propagation along linear structures and flat areas. The tuned correspondence algorithm is evaluated in terms of accuracy, robustness, and ability to generalise. Both the tuning cost function, and the evaluation are designed to balance the accuracy-robustness trade-off inherent in patch-based methods such as BFP.
Relevance and contribution: As stereo algorithms become increasingly complex, the need for automatic tuning and optimisation increases. This requires an automatic tuning and evaluation procedure tailored to the specific setting in which the method will be used. The calibration and ground-truth generation procedure using a common and inexpensive depth sensor, in combination with an accuracy/robustness balanced optimisation procedure should be of value to anyone wanting to adapt a stereo algorithm to their own needs. The extension of BFP to multiple scales is also shown to increase performance, which should be of interest to those using this type of method.

The author has contributed to the calibration and optimisation procedures required for ground-truth generation and automatic parameter tuning. The author has also contributed to the adaptation, extension and incorporation of the stereo algorithm into the pre-existing object recognition platform and the writing of the paper.

Paper E: Improving Random Forests by correlation-enhancing projections and sample-based sparse discriminant selection

Marcus Wallenberg and Per-Erik Forssén. Improving Random Forests by correlation-enhancing projections and sample-based sparse discriminant selection. In *Computer and Robot Vision*, 2016

Abstract:
Random Forests (RF) is a learning technique with very low run-time complexity. It has found a niche application in situations where input data is low-dimensional and computational performance is paramount. We wish to make RFs more useful for high dimensional problems, and to this end, we propose two extensions to RFs: Firstly, a feature selection mechanism called correlation-enhancing projections, and secondly sparse discriminant selection schemes for better accuracy and faster training. We evaluate the proposed extensions by performing age and gender estimation on the MORPH-II dataset, and demonstrate near-equal or improved estimation performance when using these extensions despite a seventy-fold reduction in the number of data dimensions.
Relevance and contribution: The ability to use lightweight random forest estimators on high-dimensional multi-parameter estimation problems should be useful to anyone wanting to deploy such an estimation system in a setting where speed and memory constraints prohibit the use of other universal estimators. The problem-aware automatic fashion in which a feature space is found is also applicable to any other estimator one might wish to train, making it a potentially useful preprocessing step in many other applications. Finally, the improved discriminant selection scheme, if properly optimised, can also be used to introduce sparsity and decrease training times in other estimators based on linear or affine components.

The author has performed the estimator implementation, and designed the structure of the feature space and discriminant selection schemes. The author has also carried out the experimental evaluation of the system and contributed to the writing of the paper.

Paper F: Attention Masking for Pre-trained Deep Networks

Marcus Wallenberg and Per-Erik Forssén. Attention Masking for Pre-trained Deep Networks. (Submitted), 2017

Abstract:
The ability to direct visual attention is an important skill for artificial recognition systems. In this paper, we study the effects of attentional masking within pre-trained deep neural networks for the purpose of handling ambiguous scenes containing multiple objects. Such situations frequently arise in the context of visual search and object recognition on robot platforms, where whole-image classification is not applicable and a full semantic segmentation is either impossible or too costly. We investigate several variants of attentional masking on task-adapted partially pre-trained deep neural networks and evaluate the effects on classification performance and sensitivity to attention mask errors in multi-object scenes. We find that a combined scheme consisting of multi-level masking and blending provides the best trade-off between classification accuracy and insensitivity to masking errors. For reasonably accurate masks it can suppress the influence of distracting objects and reach comparable classification performance to unmasked recognition in cases without distractors.
Relevance and contribution: Using attention masking to modulate an existing pre-trained deep network allows for the application of widely available pre-trained models without modifying the network structure or applying potentially destructive image operations to the input. As such, this procedure should be of interest to anyone wanting to use a pre-trained network for recognition on cluttered scenes. As the procedure can also reduce the impact of domain shift in transfer learning, it is a potentially useful add-on when problems arise due to these effects.

The author has implemented the masking methods investigated and assembled and trained the networks used. The author has also carried out the experimental evaluation on in both the dataset and robot scenarios and contributed to the writing of the paper.

Other Publications

Parts of the material presented in this thesis also appeared in the author's licentiate thesis:

> Marcus Wallenberg. *Components of Embodied Visual Object Recognition: Object Perception and Learning on a Robotic Platform*. Linköping University Electronic Press, 2013. ISBN 978-91-7519-564-3. Licentiate Thesis no. 1607.

The following other publications by the author are related to the included papers:

> Marcus Wallenberg. A Simple Single-Camera Gaze Tracker using Infrared Illumination. In *Proceedings of SSBA 2009 Symposium on Image Analysis*, pages 53–56, 2009.
> (Early work on a system for studying attention by gaze tracking)

> Marcus Wallenberg, Michael Felsberg, Per-Erik Forssén, and Babette Dellen. Leaf Segmentation using the Kinect. In *Proceedings of SSBA 2011 Symposium on Image Analysis*, 2011b.
> (Preliminary version of paper C)

> Marcus Wallenberg and Per-Erik Forssén. Automatic Stereo Tuning for YOUR Robot. In *Proceedings of SSBA 2013 Symposium on Image Analysis*, 2013.
> (Shortened version of paper D)

> Sanna Ringqvist, Pelle Carlbom, and Marcus Wallenberg. Classification of Terrain using Superpixel Segmentation and Supervised Learning. In *Proceedings of SSBA 2015 Symposium on Image Analysis*, 2015.
> (Work on an application of decision forests in which the author had an advisory role)

2

What is embodied visual object recognition?

"We're working on object recognition", I said. "Recognition? it can't be that hard, you just take two pictures and check if it's the same thing in both of them!", the visitor replied. We were at a popular science demonstration for secondary school students which had also attracted a number of members of the general public. "Very well...", I said. "How would you go about doing that?". "Well... you just... you know... compare them...", the gentleman replied, trailing off.

This brief exchange has stuck with me for the simple reason that we, as humans, have a remarkable tendency to take our ability to learn and recognise the world around us with little to no effort at all for granted. After all, a small child can easily outperform the best and latest multi-million dollar recognition systems when it comes to seeing, learning and generalising semantically relevant information about objects in the real world, so how hard could it be? Large artificial systems now readily outperform humans in highly specialised tasks such as fine-grained recognition. Even action-based tasks that require planning and learning of action relations are beginning to emerge (see for instance (Mnih et al., 2015)). However, these are usually capable only within a limited scope, far narrower than that of many tasks in a typical human's daily life.

Since biological recognition systems are both ubiquitous and remarkably successful, it is only natural to look to them when studying object recognition. There are a few traits that most (if not all) have in common. All recognition systems in nature are *embodied*, that is, they exist only within the organism they belong to and do not exist as separate entities. They are also (to varying extents) *learned*, in that they change in order to incorporate the percepts encountered during the lifetime of the organism. The EVOR project (see section 1.1.1) sought to study these kinds of mechanisms in an artificial system (see chapter 12), and to attempt to incorporate the embodiment aspect into the recognition process.

2.1 A recognition system in the real world

The utility of a biological recognition system is (from an evolutionary standpoint) the preservation and proliferation of the host organism. There are a number of recognition tasks that are useful (and often necessary) for survival. However, all recognition tasks are not created equal, and place different requirements on both the speed and nature of the recognition process, as exemplified in figure 2.1. Aspects include:

- *speed* - how long until a decision is made?
- *level of detail* - how specific does it have to be?
- *persistence* - does this affect future decisions?
- *experience* - how does this affect the concept of the outside world?
- *abstraction* - what level of generalisation from experience is needed?
- *inference* - what is the effect of previous beliefs on the current decision?

Some tasks (predator evasion, for instance) require speed, but (at least initially) little of the other aspects. Others, such as navigation, or comparison of objects and situations from memory may require both a high level of detail and persistent memory, but are not necessarily fast. The understanding of object categories, and similarities between classes of objects requires abstraction, but is learned slowly compared to specific classes. Abstractions, provided they are at least somewhat reliable, are useful for making assumptions about things that are not directly observable, but are suggested by what *can* be seen. Finally, inference regarding the properties and affordances (see (Gibson, 1977)) of novel objects is the most abstract of these since it requires thinking "outside the box", rather than relying on what is "known" [1]. Reasoning not only about the physical qualities of a novel object, but also about its potential utility is in many ways the Holy Grail of object recognition. This however (to the best of the author's knowledge), lies far beyond the capabilities of any current artificial system.

2.2 How do embodied systems learn?

Although much of what we think of as "learning" is rather "education" (that is, some kind of guided knowledge transfer), successful learning must have a perceivable effect in order to make any lasting impression. This is a consequence of the computational architecture of the brain, as learning in practice is all about adapting the neuronal structure itself (for a description of the stages of neuronal development, see for instance (Solso et al., 2008)). The general principle is that of *perception-action learning*, in which a percept triggers a response, and feedback

[1] The undisputed (albeit fictional) master of this has to be Angus MacGyver, whose very name has entered the mainstream vocabulary as a synonym for exploiting object affordances in ingenious ways.

Figure 2.1: *These situations and object require different levels of speed and specificity. Some decisions must be very specific due to the similar nature and affordances of object classes. Also required is the ability to draw conclusions about other objects (such as directions from signage), or generalisation from partial information due to occlusion.*

is obtained as the observed change in this percept or another. In an evolutionary context, the only truly permanent feedback is the survival or non-survival of the organism. However, within a social context, experience which is not encoded in organisms themselves (i.e. learned concepts), can be propagated to other individuals (i.e. supervised learning). The ultimate benefit of the perception-action learning is that it allows a principle learned through feedback and reinforcement to be applied to new situations, and thereby allow a system (biological or otherwise) to make inferences about the utility of future actions.

2.3 Building an artificial embodied recognition system

So, what is needed for learning and recognition in the case of visual object recognition? First of all, some way of *seeing*, or rather, some way of observing the world using one or more *eyes* (see chapter 3). In an artificial system, this would include some kind of imaging system (i.e. one or several cameras, see chapter 4). Functions of the data generated by the imaging system, such as spatial information from stereo (see chapter 5) and visual attention from a saliency detector (see chapter 6) also play an important part in gaze control and object discovery. If there is then to be some notion of *objects*, a mechanism by which these can be observed as separate entities (i.e. a *segmentation*, see chapter 7) is also needed. Mechanisms for extraction of information, abstraction and permanent memory (i.e. *description*, *learning* and *representation*, see chapter 8) are also required. Ideally, the system would also incorporate aspects of task specificity (such as object utility and affordances), but this lies outside the scope of this thesis. A prototype system used to investigate these aspects is described in chapter 12 and the techniques used are also the subject of the publications in part II. An interesting (and sometimes infuriating) aspect of learned systems is that, since the system structure is a result of the learning process, analysis is often difficult. A system may work (or not), while providing very little apparent information about *why*

this is the case. As learning systems grow in complexity, this problem increases in severity until systematic construction and design is impossible. At this point (as is still the case with, for instance, parts of the human brain), the system becomes nothing more than a "magic black box". Thus, research into analysis tools for learning systems is also an important emerging area related to recognition.

3

Eyes and vision

Eyes are (quite obviously) necessary for vision. Eyes also come in many different shapes and sizes, and differ from each other in many respects (some examples are shown in figure 3.1). However, the optical principles underlying image formation are universal and, as such, govern the principles by which eyes can be constructed. In this chapter one particular type of eye, the camera-type (chambered) eye of humans, will be briefly illustrated and some of its characteristics vital to visual search and object recognition described.

3.1 The camera-type eye

The camera-type eye, consisting of a chamber, an aperture (*pupil*) and a photosensitive surface (the *retina*) is one of the most common kinds of eye in the animal kingdom. Many of these also include a system for focussing light onto the retina. In the case of the human eye this is accomplished using the *cornea* for coarse focus and the *lens* for finer control. A schematic view of this can be found in figure 3.2. Incident light from the surroundings passes through the cornea, lens and pupil and onto the retina. There, cells containing light-sensitive *opsins* (substances belonging to the G protein coupled receptors) react by emitting voltage spikes at a rate inversely proportional to the intensity of the incident light. Through a series of neural layers composed of different cell types with varying receptive fields, the retina is connected to the optic nerve, which in turn leads to the *lateral geniculate nuclei* and onward into the brain[1].

[1] For a more in-depth description of the biology of the human visual system, see for instance (Stone, 2012)

Figure 3.1: *Different kinds of eyes. From left to right: a human eye (the author's), a cat's eye (another mammalian camera-type eye), the three pairs of different eyes of a jumping spider (family Salticidae), compound eyes of Meligethes aeneus (a pollen beetle).*

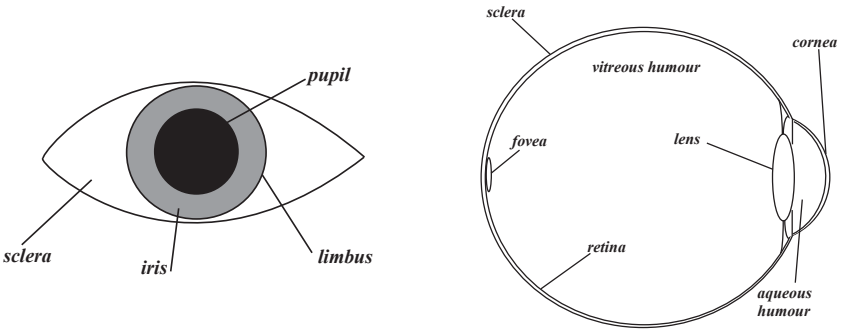

Figure 3.2: *Schematic views of the human eye. Frontal view displayed as occluded by eyelids. Note the small extent of the fovea on the retinal surface, limiting high-acuity vision to approximately 2°.*

3.2 Peripheral and foveal vision

In humans (to name but one example), the distribution of photoreceptor cells across the retina is far from uniform. The total density of receptor cells falls off sharply around a high-density region known as the *fovea*[2]. The consequence of this is that visual acuity decreases rapidly when moving away from this region. The high-acuity fovea accounts for approximately 2° of the 140° visual field of the eye. An illustration of this (not to scale) of this can be seen in figure 3.3.

[2]This is especially true for the so-called *cone*-type cells responsible for chromatic vision, and less so for the *rod*-type cells used for achromatic vision as they are almost absent within the fovea. The statement concerns the combined photoreceptor density, regardless of chromatic sensitivity.

Figure 3.3: *Illustration of varying visual acuity. Left: low-acuity image representing peripheral vision. Centre: high-acuity image representing foveal vision. Right: radially symmetric logarithmic transition from high to low acuity around the image centre.*

3.3 Saccades and fixations

In conjunction with the non-uniform acuity of many vision systems, there exists in most visually capable species a motor control system for moving either the eye (as in humans), the head (as in birds) or sometimes the entire body (as in certain insects) of the organism to align high-acuity vision with a specific spatial location[3]. These movements are known as *saccades*, and they typically move from location to location in a *saccade-and-fixate* pattern. These *fixations* can occur in very rapid succession (in humans, usually 3-4 times per second), making high-acuity vision available in almost any region of the visual field at a moment's notice. Although they are referred to as fixations, this does not imply that the eye is fixed. In humans for instance, during fixation, there is still a small, tremor-like motion of the eye known as *microsaccadic* motion. While hypotheses on the reasons for (and purposes of) this motion, it is not yet well understood. Also, fixation does not necessarily relate to a fixed spatial position, as many organisms can perform the same saccade and fixation pattern while attending to a moving target and stabilising the retinal image using a *pursuit*-type motion. For a more detailed description of saccadic motion, see for instance (Land and Nilsson, 2002). The saccade and fixation phenomena are closely coupled with the notion of *visual attention*, which will be explored further in chapter 6.

[3]The "chicken and egg" problem of the co-evolution of visuomotor control and non-uniform visual acuity is left unexamined here. Discussions of this can be found in for instance (Nilsson, 2009).

3.4 Spatial information, vergence and parallax

In vision systems with either at least two eyes and overlapping visual fields, or the ability to change viewpoint, it is possible to extract spatial information through either stereo vision, motion parallax or a combination of both. In humans, both of these are possible, since we are equipped with two forward-facing eyes with overlapping visual fields and have the ability to move in order to create motion parallax. In addition to this, there exists (not only in humans) a vergence system capable of orienting the eyes such that binocular high-acuity foveal vision is achieved at or around a certain location in three-dimensional space. This range accommodation by eye vergence is necessary for focussing on objects and aligning the receptive fields of both eyes for binocular vision. At greater range, monocular range cues such as perspective effects and assumed scale play a greater part than binocular ones and vergence is less apparent. The concept of spatial information from parallax (or *disparity*) in artificial systems, is described further in chapter 5.

4

Single and multiple view geometry

A *camera* (latin for "chamber"), in the broadest sense of the word, is a device which creates images of the world through some form of projection. Examples of such cameras range from the venerable *camera obscura*, which may consist of nothing more than a box with a pinhole, to the camera-type eye of many animals (such as humans, as illustrated in the previous chapter) and the digital cameras of today. Modelling these systems is necessary in order to describe the process of image formation, and to understand the relationship between images and the real world. This chapter will give a brief introduction to common camera models, and their application to single and multiple view geometry.

Figure 4.1: *Cameras come in many shapes and sizes. From left to right: a simple pinhole camera constructed from a cardboard box, a compact 35 mm camera, a digital camcorder, a camera-equipped mobile phone (now the world's most common camera system).*

4.1 Single-view geometry and the pinhole camera

The pinhole camera is the most basic of camera models, being both the first described (at least as early as the 4th century B.C. (Campbell, 2005)) and the first to see practical use. The ideal pinhole camera consists of an aperture of zero width, through which light from the surroundings is projected. Geometrically, this can be described in a simple way. The most common formulation is as follows: Assuming the origin at the camera aperture and an image plane orthogonal to the optical axis of the pinhole camera, at a distance d from the aperture, the relationship between a point $\mathbf{x}$ in $\mathbb{R}^3$, as described by the homogeneous coordinates $\mathbf{x} = [x, y, z, 1]^T$ and its image $\mathbf{x}_p = \left[x_p, y_p, d\right]$ can be expressed using the projection mapping $\mathbf{C}$ as

$$\mathbf{x}_p \sim \mathbf{C}\mathbf{x}, \tag{4.1}$$

where $\sim$ denotes projective equivalence. Assuming the coordinates of both $\mathbf{x}$ and $\mathbf{x}_p$ are expressed in a right-handed orthonormal coordinate system,

$$\alpha\mathbf{x}_p = \mathbf{C}\mathbf{x} = \begin{bmatrix} \alpha x_p \\ \alpha y_p \\ \alpha d \end{bmatrix}, \tag{4.2}$$

where α is an arbitrary scaling factor. The camera-centered coordinates of $\mathbf{x}_p$ can then be obtained by dividing by α.

For practical reasons, the distance d to the image plane is often normalised to 1, corresponding to what is commonly known as the *normalised image plane*, where a point is imaged as

$$\mathbf{x}_n = \left[x_n, \quad y_n, \quad 1\right]^T = \left[\tfrac{1}{d}x_p, \quad \tfrac{1}{d}y_p, \quad 1\right]^T. \tag{4.3}$$

The camera matrix $\mathbf{C}$ describes the camera position and orientation (relative to some origin in the world) such that

$$\mathbf{C} = \left[\mathbf{R}^T, \quad -\mathbf{R}^T\mathbf{t}\right], \tag{4.4}$$

where $\mathbf{R}$ and $\mathbf{t}$ are a rotation matrix and a translation vector, respectively. Usually, these are collectively known as the *extrinsic camera parameters*.

4.1.1 Thin-lens cameras and their pinhole approximations

For practical purposes however, actual pinhole cameras are most often insufficient. Almost all modern cameras rely on lenses to gather and focus light at a focal point, necessitating a slight change to the "pure" pinhole camera model. Assuming a thin rectilinear lens[1], the pinhole camera model can be used to describe the image of "distant" points. The optical centre of the lens replaces the aperture of the pinhole camera as the origin, but is otherwise equivalent, and the distance d from the aperture is replaced by the focal distance f of the lens.

[1] That is, a lens through which lines in the world project to lines in the image.

This models the thin lens camera with focal length f as a pinhole camera with an image plane distance of f according to figure 4.2. The imaged point $\mathbf{x}_p$ above is then $f\mathbf{x}_n = [fx_n, fy_n, f]^T$ in world coordinates (the normalised image plane coordinated however, remain unchanged).

Clearly, this model holds only for a specific distance between the imaged point and the camera, resulting in image blur when this condition is not met[2]. However, since the pinhole camera approximation provides such an algebraically simple model, it will henceforth be used for all cameras in this thesis.

4.1.2 Digital cameras and the pixel grid

For obvious reasons, the images encountered in digital image processing are without exception[3] captured by projection onto an electronic sensor grid. This adds yet another aspect to the imaging, namely the relation between the projected point $\mathbf{x}_n$ in the normalised image plane and the homogeneous coordinates of its corresponding pixel location $\mathbf{u} = [u, v, 1]^T$. Usually, this relation is described by a linear mapping $\mathbf{K}$ from the normalised image plane to the pixel grid, where

$$\mathbf{u} = \mathbf{K}\mathbf{x}_n = \begin{bmatrix} f_u & \gamma & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_n. \tag{4.5}$$

The f_u and f_v parameters describe the relation between world distances and distances along the u and v axes of the pixel grid. The γ parameter determines the skew of the pixels relative to the v axis, and c_u and c_v define the origin in the pixel grid.

With this in place, the mapping from a world point to a pixel location under this camera model can be formulated as

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} \alpha f u \\ \alpha f v \\ \alpha f \end{bmatrix} = \mathbf{K}\mathbf{C}\mathbf{x}, \tag{4.6}$$

and finally,

$$\mathbf{u} = \frac{1}{\alpha f}\mathbf{K}\mathbf{C}\mathbf{x}. \tag{4.7}$$

[2] The distance range producing acceptable sharpness is known as the *depth of field* of the camera.
[3] In the sense that the author is unaware of any exceptions.

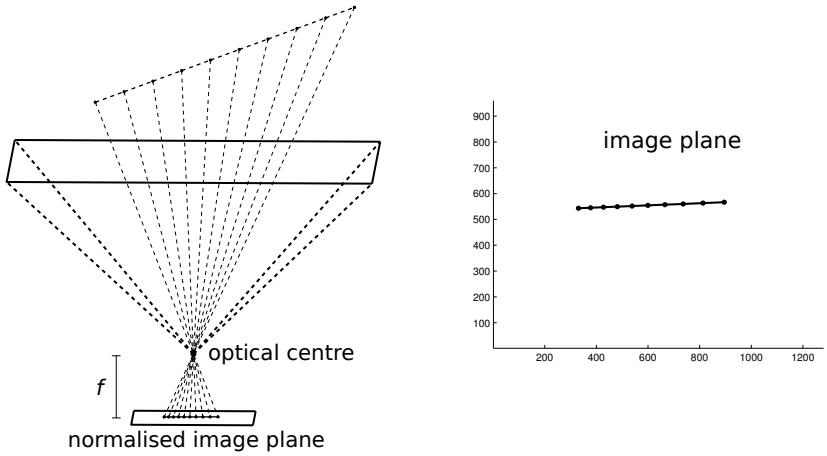

Figure 4.2: *Single camera geometry. Left: projection of points along a line in space to the normalised image plane of a camera. Shown also is the focal length, denoted f. Right: image locations of the resulting projected points on the pixel grid.*

4.2 The effects of lens distortion

The camera model in section 4.1 assumes that the projection of points is rectilinear. This is a necessary assumption if the whole projection is to be modelled by a single linear operation on the homogeneous coordinates of a point in the world. However, this is seldom the case in practice, since the thin lens camera model is not satisfied by real-world lenses. These effects can be mitigated by using multiple lenses to achieve an approximately rectified image, but some effects usually remain.

4.2.1 Common types of lens distortion

There are several common types of lens distortion effects, caused by different lens types, and different manufacturing processes. However, they all stem from an uneven refraction of incoming light across the lens. The most commonly encountered are the *barrel* and *pinchushion*-type distortions, or a combination of both (sometimes called a *moustache* distortion, due to its effect on horizontal lines). Examples of these can be seen in figure 4.3.

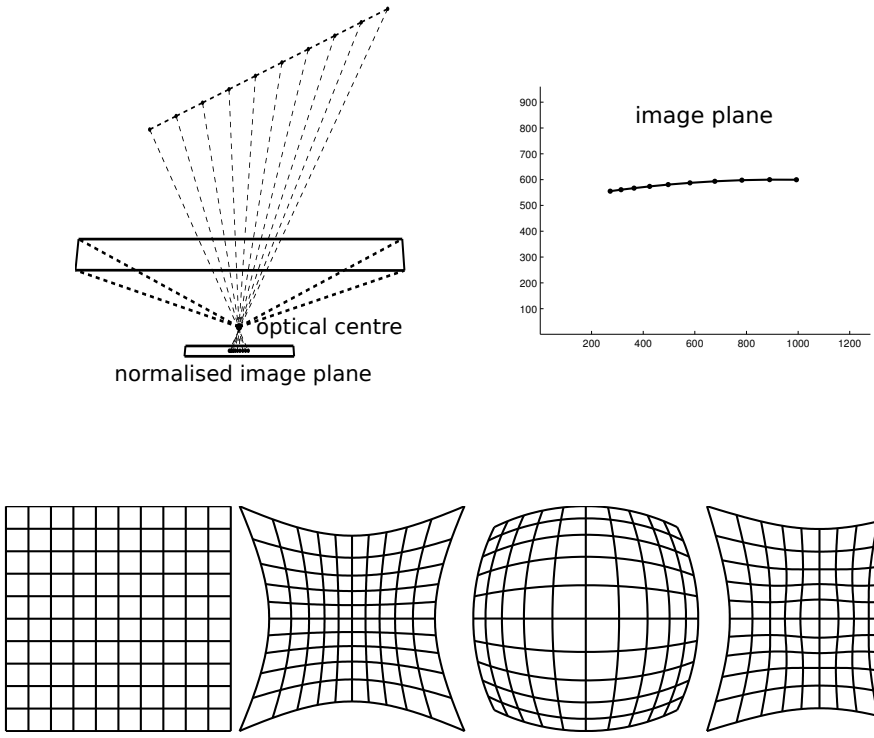

Figure 4.3: *Single camera geometry under lens distortion. Top left: projection of points along a line in space under barrel-type distortion. Top right: resulting image plane locations (notice the curvature of the projected line due to lens distortion). Bottom row: examples of distortion types. From left to right: undistorted grid, pincushion distortion, barrel distortion, moustache distortion.*

4.2.2 Lens distortion in the single-camera case

In the single-camera case, the effects of lens distortion are most commonly modelled in the normalised image plane (i.e. the normalised projection through the lens onto the sensor). Some distortion models operate directly on the final image (such as the model used for wide-angle lenses in (Wallenberg and Forssén, 2012), see part II, paper D). However, this requires the lens distortion model to take into account pixel aspect ratio and skew as well as effects of the actual lens. In the former case, the lens distortion effects can be incorporated into the pinhole camera model through the use of an invertible lens distortion function $\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ such

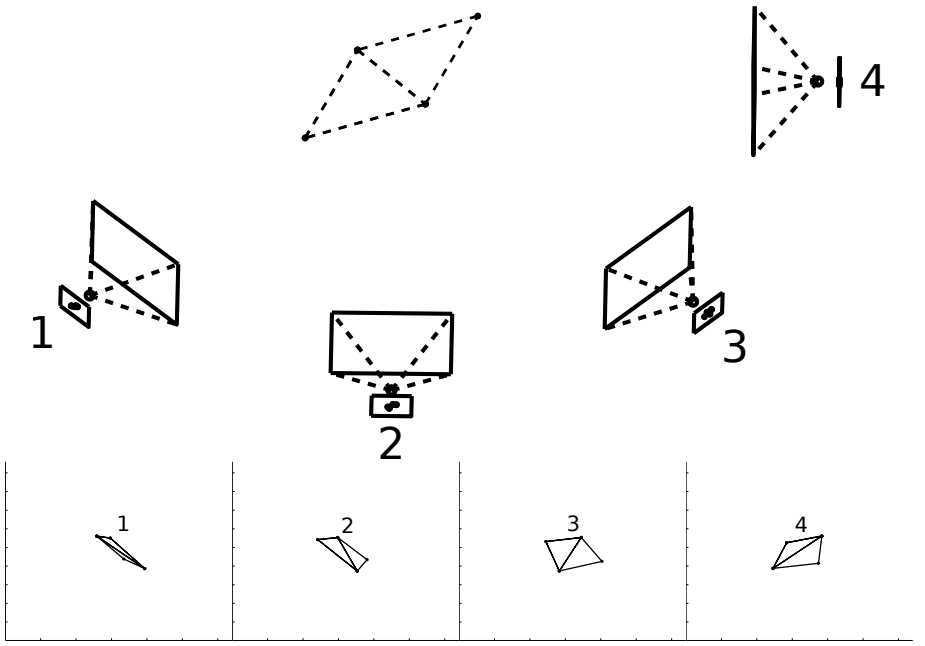

Figure 4.4: *Multiple view geometry. Projection of points on a plane to four cameras at different positions. Top: 3D positions of the points and cameras. Bottom: resulting image plane projections of the respective cameras.*

that

$$\mathbf{x}_n = \mathbf{f}^{-1}\left[\tilde{\mathbf{x}}_n\right], \tag{4.8}$$

in equation (4.5), where $\tilde{\mathbf{x}}_n$ represents the image of the world point $\mathbf{X}$ under lens distortion and $\mathbf{x}_n$ the compensated rectilinear projection[4]. Thus, the complete transformation from world to image becomes

$$\mathbf{u} = \mathbf{K}\mathbf{f}\left[\mathbf{p}\left(\mathbf{C}\mathbf{x}\right)\right], \tag{4.9}$$

where $\mathbf{p}\left(\right)$ denotes the projection to the normalised image plane (as done in (4.3)) such that

$$\tilde{\mathbf{x}}_n = \mathbf{p}\left(\mathbf{x}_p\right) = \mathbf{p}\left(\mathbf{C}\mathbf{x}\right). \tag{4.10}$$

If the lens distortion is instead applied to the actual (non-normalised) image plane, the order of the mappings $\mathbf{K}$ and $\mathbf{f}\left(\right)$ are reversed.

[4]Since the distortion function usually operates only within a plane, it can be (and often is) envisioned as a mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^2$. However, since we describe projected points as vectors in $\mathbb{R}^3$, this formulation makes for simpler expressions.

4.3 Multiple view geometry

In many situations, it is necessary to consider not only a single camera, but multiple ones. This can mean either several physical cameras, or several "virtual cameras", corresponding to a single physical camera at different points in time, at different locations and/or with different parameters. Assuming the pinhole camera model (including lens distortion), each camera can be modelled as in (4.9). An illustration of this can be seen in figure 4.4. If the world positions of points are known, correspondences can then be established across multiple views. The image of a point in multiple cameras can then be determined from the location, orientation and intrinsic parameters of these cameras.

Conversely, the location of a point in the world can be determined from at least two of its projections if the extrinsic, intrinsic (and distortion) parameters of the corresponding cameras are known. The fixed two-camera case is discussed further in chapter 5. If the camera locations and orientations are unknown, the relative camera and point locations can still be estimated up to scale. With multiple points and cameras, this results in the extensively studied *bundle adjustment* problem (Triggs et al., 2000), where both camera parameters and point locations are jointly optimised.

5

Stereo Vision

Stereopsis, the extraction of spatial information in three-dimensions using binocular vision, is a very useful tool in both natural and artificial vision systems. Since the advent of stereograms in the 19th century, the mechanisms by which depth information is extracted from binocular cues have been the topic of much research. The principles of two-view geometry have also become a cornerstone of computer vision. Although the mechanisms by which biological systems accomplish this are not fully understood, the requirements for recovery of 3D structure from two views are. This chapter will describe a typical stereo vision setup, as well as methods for correspondence estimation and reconstruction. The special case of the *structured light* technique for aiding correspondence estimation will also be described.

5.1 Epipolar geometry

Typically, when first explaining stereo photography, most sources describe the case of two pinhole cameras imaging a single point in space. The image of this point in one of these cameras is formed by projection to the image along a single ray extending through the point in space and the optical center of the camera. Under the same pinhole camera assumptions as in (4.7), it is evident that all points along this reprojection line correspond to the same location in that specific image. When viewed from the other camera, however, points at different distances along the line project on a line in the image. Moreover, all such lines converge on the image of the first camera center, known as the *epipole*. Thus, for each possible location in one of the images, there exists in the other an *epipolar line*, on which all possible world points projecting to this location are imaged, as illustrated in figure 5.1. This constraint is commonly known as the *fundamental matrix constraint*,

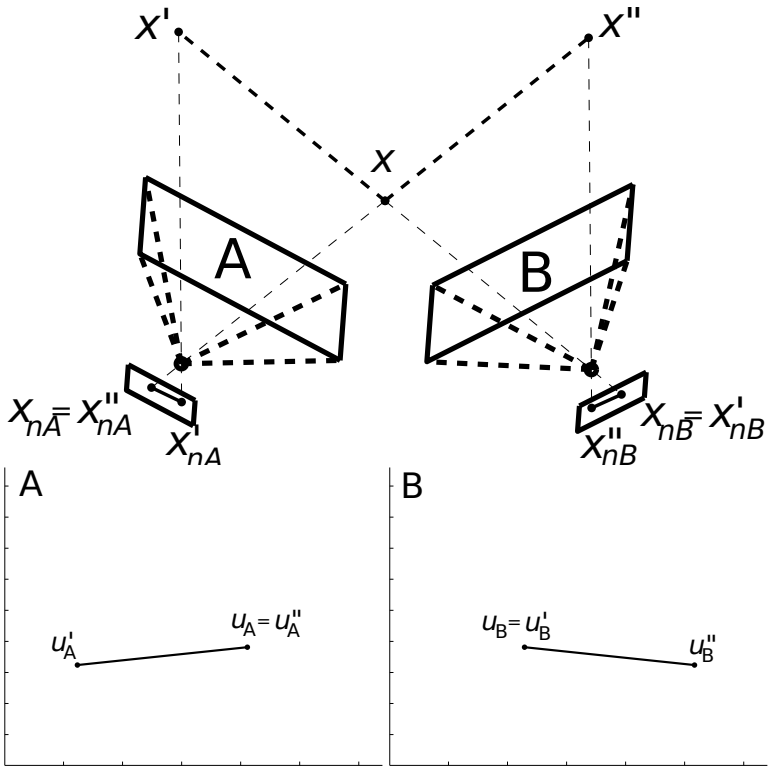

Figure 5.1: *Rectilinear epipolar geometry. Top: 3D positions of cameras A and B. Points* $\mathbf{x}$ *and* $\mathbf{x}$*" project to the same normalised image plane location* $\mathbf{x}_{nA}$ *in camera A. Points* $\mathbf{x}$ *and* $\mathbf{x}'$ *project to the same normalised image plane location* $\mathbf{x}_{nB}$ *in camera B. Bottom row: corresponding image plane projections. The image plane disparities between pairs of corresponding points can be defined as* $\mathbf{d} = \mathbf{u}_B - \mathbf{u}_B$,$\mathbf{d}' = \mathbf{u}'_B - \mathbf{u}'_B$ *and* $\mathbf{d}'' = \mathbf{u}_B'' - \mathbf{u}_A''$.

and expressed by means of a 3×3 *fundamental matrix* $\mathbf{F}$ as

$$\mathbf{u}_A^T \mathbf{F}_{AB} \mathbf{u}_B = 0, \tag{5.1}$$

where $\mathbf{u}_A$ and $\mathbf{u}_B$ are projections of the same world point to cameras A and B, respectively, and $\mathbf{F}_{AB}$ is a fundamental matrix relating these cameras.

The magnitude and direction of the image-plane offset between the projections $\mathbf{u}_A$ and $\mathbf{u}_B$ (commonly known as a *disparity*), is then dependent on the distance of the imaged point from the baseline (the line connecting the two camera centers). In practice this means that, if the epipolar geometry is known, the set of possible image locations of corresponding points can be determined. As mentioned in section 4.3, the world location of a point can be determined from two such images by means of triangulation. The largest part of stereo vision

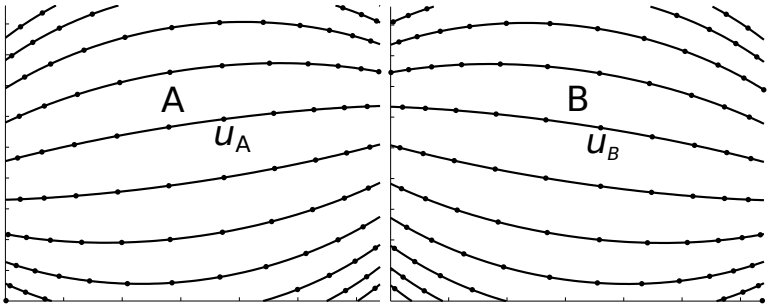

Figure 5.2: *Epipolar curves under distortion. Corresponding projections* $\mathbf{u}_A$ *and* $\mathbf{u}_B$ *in the image planes of cameras A and B respectively. Other points illustrate epipolar lines, curved by lens distortion of the barrel type.*

research is concerned with this correspondence estimation and recovery of three-dimensional structures. What makes the rectified case so attractive is that, with known epipolar geometry, this can be simplified to a correspondence search along an epipolar line.

Another advantage of the rectified case is that, for every pair of images from such a stereo pair, there exists a non-unique set of *rectifying homographies* that transforms the images such that the epipolar lines become parallel (Loop and Zhang, 1999). This means that all disparity vectors are also parallel, and that correspondence estimation reduces to a set of searches along parallel lines. The rectifying homographies are often chosen such that the epipoles are placed at infinite distance from the camera centers on the horizontal axis, meaning that correspondences can be searched for along pixel rows. An in-depth description of the details of epipolar geometry can be found in (Hartley and Zisserman, 2000).

5.1.1 Epipolar geometry and lens distortion

While the rectified stereo case is well-studied, it relies on a number of assumptions that are rarely satisfied by real-world camera setups. Especially in the case of wide-angle imagery, the effects of lens distortion (such as those illustrated in section 4.2.2) mean that the epipolar geometry cannot be expressed using epipolar lines, rectifying homographies, and a fundamental matrix[1]. An example of this is shown in figure 5.2. In order to make use of the theory of rectified epipolar geometry, many methods therefore include distortion compensation and rectification as a preprocessing step. However, this means that any errors introduced by these will carry over and affect the final correspondence estimation.

[1] It *is* possible to formulate this using entities akin to these. However, the algebraic advantages of describing them using point-line interactions are lost, and the parameterisation becomes more complicated.

5.2 Stereo vision algorithms

The aim of all stereo vision algorithms is to establish correspondences between two images. Depending on the assumptions made and the techniques employed, they can be roughly divided into two categories: *global* methods that match entire images under various constraints, and *local* methods that are only concerned with a small neighbourhood around the points for which correspondences are to be determined. There are also methods that do not belong entirely to either category, such as *correspondence propagation*, which will be described further below.

5.2.1 Global versus local methods

The simplest kind of stereo correspondence algorithms work by matching small neighbourhoods of points independently of each other. For instance, each $M \times N$ pixel patch in an image may be correlated with every $M \times N$ pixel patch in another image, and the best match chosen as the estimated correspondence. This kind of unconstrained matching is rarely performed in practice, due to the computational cost of an exhaustive search, and the lack of any enforced consistency. Typically, local methods are instead restricted to evaluating potential matches within a neighbourhood defined by a disparity limit and (often) the estimated rectified epipolar geometry. In this way, the search can be restricted to evaluating a small number of possible correspondences on or around an epipolar line. It is also common to apply such methods in a *coarse-to-fine* manner, in which an initial result at coarse scale is refined using a progressively more restricted search space at finer scales.

Global methods typically rely on both local comparisons and a global cost function based on the smoothness and consistency of the estimated disparity field. A common approach is to view the disparity as the result of a stochastic process and combine the local similarity and the global smoothness and consistency in a random field, which is then optimised using energy minimisation techniques.

Purely local methods tend to produce a noisier result due to the lack of enforced consistency, but are also unaffected by the over-smoothing that regularisation of the disparity estimate can bring. Also, there are typically gaps in the estimated disparity due to lack of texture or due to occlusion. Global methods in general produce a dense and smooth result, due to the regularisation applied in the estimation process. While these methods can produce estimates in untextured or occluded areas by extrapolation based on neighbouring regions, this can also lead to incorrect estimates and over-smoothing of fine details. For a more in-depth discussion of the components of stereo algorithms, see for instance (Scharstein and Szeliski, 2002).

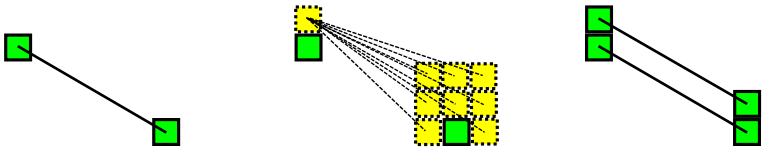

Figure 5.3: *Correspondence propagation. Left: a pair of seed correspondences established by initialisation procedure. Centre: potential correspondences for a neighbour of one of the seed points (dashed). Right: best correspondence found after evaluating neighbours, added as a new seed correspondence.*

5.2.2 Correspondence propagation

The idea of correspondence propagation is to make use of an implicit smoothness constraint by observing that the disparity values of neighbouring image locations are often highly correlated (since they often depict points on the same surface in the world). This suggests that if a reliable correspondence can be established somehow, a reasonable assumption is that neighbouring locations will differ only slightly in disparity. Thus, a good strategy for a correspondence search is to search neighbouring locations, starting with the disparity estimate propagated from the previously matched location. This propagation can then be repeated until no further correspondences can be found. An illustration of one such step can be seen in figure 5.3.

An example of this procedure is the *best-first propagation* (BFP) algorithm described in (Lhuillier and Quan, 2002), wherein a sparse set of correspondences established by matching SIFT descriptors (Lowe, 1999, 2004) is propagated to establish a quasi-dense disparity estimate. The algorithm propagates disparity estimates by searching among neighbours of established correspondences using *zero-mean normalised cross-correlation* (ZNCC)[2] and an implicit limit on the change in disparity between neighbouring pixels determined by the size of the search area. Correspondences found with high enough ZNCC coefficients are then added to the set of established correspondences. An important feature of the method is that correspondences are propagated in *best-first* order, according to their ZNCC coefficient values. This means that more reliable correspondences will be propagated to a greater extent than less reliable ones. Thus, errors in the original sparse correspondence set are less likely to spread, since consistent reliable matches are unlikely in neighbouring areas. Sub-pixel refinement and consistent epipolar geometry can also be incorporated into the method, as is done in (Lhuillier and Quan, 2003) and (Lhuillier and Quan, 2005).

[2] Also known as the Pearson product-moment correlation coefficient.

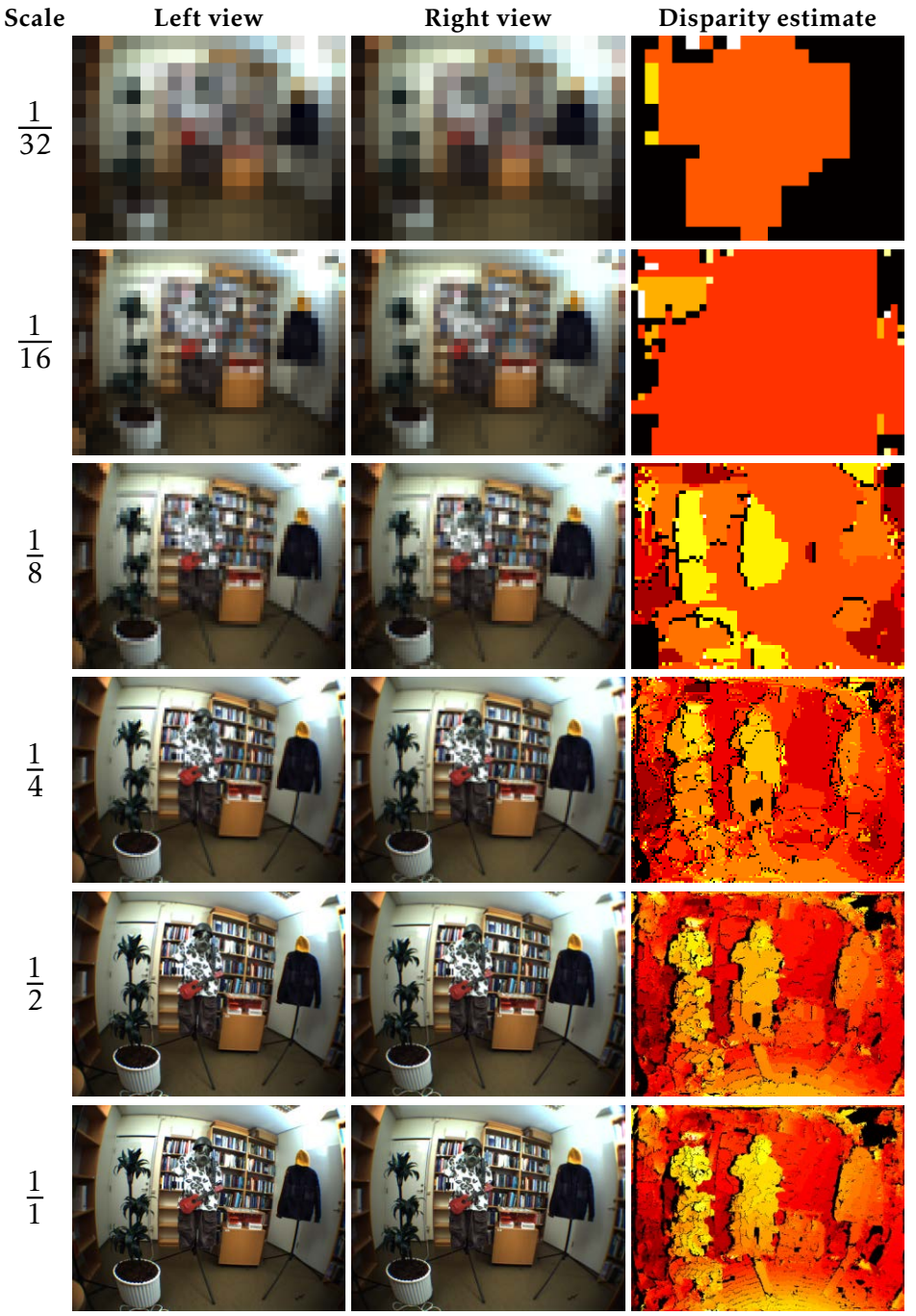

Figure 5.4: *Coarse-to-Fine Best-First Propagation. Illustration of the propagation stages of CtF-BFP. Left column: left image. Centre column: right image. Right column: magnitude of disparity estimate. Top to bottom shows intermediate results of propagation at progressively finer scales.*

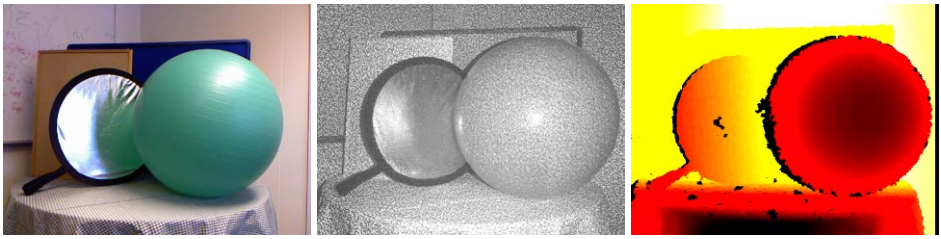

Figure 5.5: *Examples of images from a structured light system (the Microsoft Kinect). From left to right: colour image, NIR image showing structured light pattern, inverse depth image.*

5.2.3 Coarse-to-Fine Best-First Propagation

The idea behind *coarse-to-fine best-first propagation* (CtF-BFP), as described in (Wallenberg and Forssén, 2012) (see part II, paper D), is to extend the BFP algorithm by incorporating propagation across multiple scales, while limiting propagation in areas prone to drift due to the aperture problem[3]. The basic assumption in the coarse-to-fine extension is that, with sufficient sub-sampling, the disparity field at coarse scale becomes an identity mapping at the pixel level (that is, every pixel in one image corresponds to the same pixel in the other image). This provides a dense correspondence initialisation and eliminates the need for an auxiliary method of establishing seed correspondences. Starting from this, correspondences can then be propagated to progressively finer scales, refining and propagating correspondences in the image plane at each scale. An illustration of the propagation procedure can be seen in figure 5.4. The selective propagation is controlled by the degree of correlation (the ZNCC value) and by an estimate of the local intrinsic dimensionality of the imaged structure, calculated from the eigenvalues of the local autocovariance matrix.

5.3 Structured light systems

The correspondence estimation methods described above all rely on image structure (with or without some form of regularisation) to find correspondences. This means that if these structures are not visible, correspondence estimation will inevitably fail. So called *structured light* systems address this problem by providing image structures through projection of a *structured light pattern* (SLP) onto the scene. This provides the structures necessary for correspondence estimation even in the absence of object texture or external lighting. If the position of the projector and the pattern are known, the image of the SLP from the projector's point of view is also known. Thus, the projector effectively replaces one of the cameras in a stereo camera system. Moreover, if the pattern is properly designed, both the

[3]The aperture problem refers to the position and motion ambiguity inherent in certain structures when regarding only a local neighbourhood.

speed and robustness of the correspondence estimation can be improved. Structured light patterns vary in design, from line and grid patterns to dot patterns designed specifically for uniqueness and ease of matching through optimisation of their spatial arrangement[4]. Structured light systems have in recent years become commonplace due to the release of the Microsoft Kinect[5] gaming peripheral, arguably the first mass-marketed consumer-level structured light range sensor. The availability and low cost of this sensor has lead to it being applied to numerous computer vision problems related to, for instance, navigation, reconstruction and interface design. The Kinect uses an SLP in the form of a specialised dot pattern which is projected onto the scene in the *near infrared* (NIR) band. Correspondence estimation is then carried out, and the resulting disparity (in an encoded form known somewhat inappropriately as the *inverse depth map*) is returned by the device[6]. Examples of the Kinect's SLP, NIR image and the resulting inverse depth map can be seen in figure 5.5.

[4] See for instance US patent No. 20100199228A1.

[5] `http://www.xbox.com/Kinect`

[6] Since its release, several high-level libraries have been implemented to access the data in other forms. However, the basic functionality of the device remains the same.

6

Visual attention

The saccade-and-fixate behaviour of many eyes (see chapter 3) is practically useful only when employed in combination with an attention system to guide its motion. This visual attention system determines where and when the eyes should be reoriented to try to comprehend the world around them. This enables sequential visual search and scene analysis, and allows for prioritisation of potentially important parts of the visual field. In this chapter, examples of visual attention in artificial systems are discussed, along with the concepts of visual saliency and inhibition-of-return mechanisms.

6.1 What to look at

The question of what to look at cannot be answered in a straightforward manner. Visual attention in biological systems is controlled by a number of cues based on visual input, expectations, intentions and memory. While these mechanisms are not fully understood, their functions are necessary also in artificial systems and must be modelled for artificial visual search. A common model of visual attention is composed of two components, a feed-forward bottom-up component (visual saliency), modulated by a top-down *visual search* component, which is most often task-specific in nature.

6.1.1 The concept of visual saliency

Saliency (the term *salience* is more common within neuroscience) is the quality of an observation that makes it perceptually different from its surroundings in space, time or other aspects. It is also related to visual attention, since objects or structures perceived as differing from their surroundings typically elicit an attentional response in the form of visual fixation. This is commonly known as the "pop-out" effect (Treisman and Gelade, 1980), examples of which can be

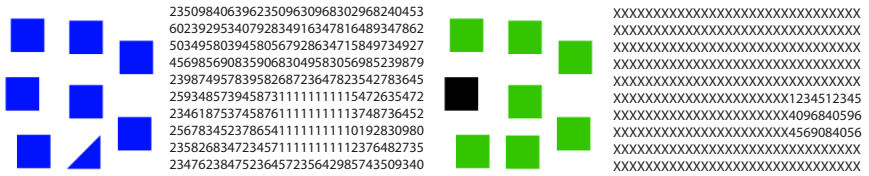

Figure 6.1: *Examples of the pop-out effect of irregularities in shape, texture and colour. From left to right: deviation in shape, a regular pattern within an irregular one, deviation in colour, disruption of a regular pattern.*

seen in figure 6.1. This effect implies that there exists a feed-forward component of visual attention closely related to these "unexpected" irregularities and that if these differences could be calculated, the feed-forward component of visual attention could be modelled.

6.2 Algorithms for saliency detection

In line with the "difference from surroundings" definition of saliency in the previous section, saliency detection algorithms are typically operations that output *centre-surround differences*, i.e. the difference between a central location, where the presence of visual saliency is to be detected, and its neighbourhood or surround. What differs between methods is how the centre and surround are defined, and in which space the difference operation is carried out. The very well-known model described by Itti, Koch and Niebur in (Itti et al., 1998) relies on extracting luminance, chrominance and orientation components from an image, computing centre-surround difference for each of these features at multiple scales, and then combining these responses into a saliency map of the image. This type of architecture is inspired by structures present in biological systems, such as the human primary visual cortex. An example of a purely colour- and intensity-based method is the *maximum symmetric surround* difference described in (Achanta and Süsstrunk, 2010), where a central pixel and the average pixel values of the largest possible symmetric neighbourhood are compared. A different approach proposed in (Hou and Zhang, 2007), instead works in the frequency domain, and calculates centre-surround differences in the log-amplitude spectrum of the image. Yet another example (Hou and Zhang, 2008) is based on the principles of image coding, and compares normalised responses to learned image features at the centre location to those encountered elsewhere in the image. There are of course many other methods that all aim to detect salient image locations. The important thing to note is that, although the specifics of these methods vary, they all rely on using information from one or more images to describe the statistics of a "typical" image region, and then compare a query location to this. In recent years, saliency operators are increasingly learned from data using machine learning methods (see for instance (Wang et al., 2016)), which places less emphasis on an explicit centre-surround

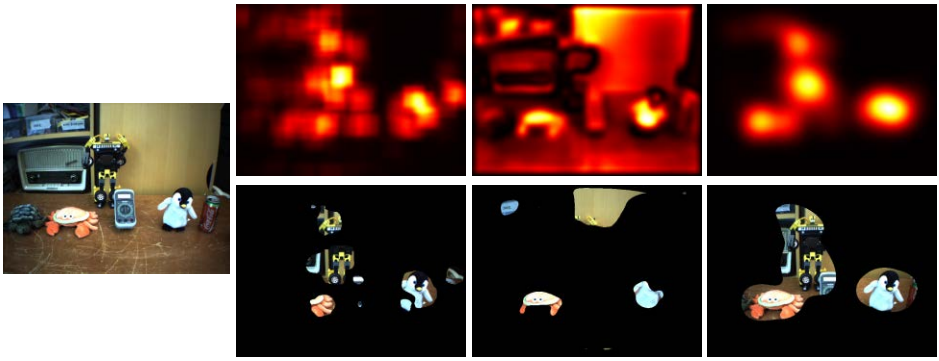

Figure 6.2: *Top row: output of some examples of visual saliency detectors. Bottom row: regions around saliency maxima obtained by automatic thresholding using the* `ocrselectthresh2` *function in* MATLAB*.. From left to right: original image, maximum symmetric surround (Achanta and Süsstrunk, 2010), spectral residual (Hou and Zhang, 2007), incremental coding length (Hou and Zhang, 2008).*

difference, but rather infers saliency directly from image content.

The final test of such methods is usually their correlation with fixation locations of human subjects, although one could argue that this fails to take into account the inevitable top-down and task-related effects encountered in human visual search[1]. For some examples of saliency maps generated using various kinds of centre-surround differences, see figure 6.2.

6.3 Dynamic visual attention and inhibition of return

Due to the inhomogeneous visual acuity in many biological systems, visual search in these systems is necessarily an active dynamic process. If visual attention were purely feed-forward, visual search would be impossible since the visual saliency of a location would depend only on the visual stimulus, and thus cause a "lock-on" effect. An important mechanism is thus the *inhibition of return* (IoR) mechanism, that modulates the bottom-up component of attention, and shifts gaze from location to location. The temporal aspect of visual attention is also an important part of many artificial attention systems, and several techniques for modelling visual search and IoR exist.

For instance, in (Itti et al., 1998), IoR is described in the image plane, by suppressing saliency detections in regions of previous maxima. Thus, each time a saliency maximum is detected and "attended to", it is then removed from the

[1] For a review of visual search experiments related to control of actions, see for instance (Land, 2006).

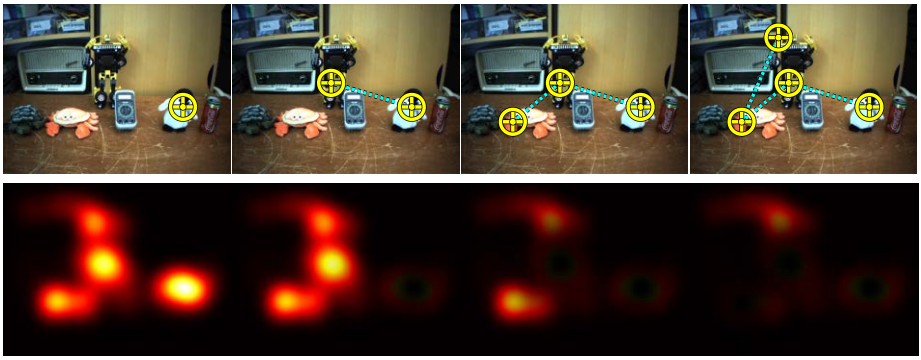

Figure 6.3: *Visual search and inhibition of return. The visual search pattern obtained by selecting saliency maxima produced by incremental coding length (Hou and Zhang, 2008) and applying image plane suppression after each target by down-weighting using Gaussian weights.*

possible candidates for fixation. In (Hou and Zhang, 2008), the IoR component is incorporated by updating the weights of features used for saliency detection according to what is encountered during the detection process. This results in a shift of the saliency maximum, as the influence of features that previously elicited the strongest response will be down-weighted over time. The IoR mechanism used in (Wallenberg and Forssén, 2010b), (Wallenberg and Forssén, 2010a) (see part II, papers A and B) is similar to the former, but instead relies on suppressing previously attended locations in the three-dimensional visual field around the platform (see chapter 12 for details). This is necessary since the part of the scene visible at any given time depends on the camera poses, and thus changes during saccadic motion. This also allows objects not currently in view to influence view planning and visual search. Examples of visual search using the image-plane suppression IoR method can be seen in figure 6.3. Combined approaches performing dynamic gaze prediction (see for instance (Borji et al., 2014)) are also of increasing interest. Such approaches combine multiple cues from image data, bottom-up saliency and other sources (such as human actions) to perform task-specific gaze prediction over time in complex scenes.

7

Segmentation

Segmentation, the act of partitioning data into disjoint subsets, is a problem that arises in all disciplines where a distinction between categories of data points is needed. Image segmentation (subdividing an image into disjoint regions), is one of the classical areas of image processing. Depending on the task at hand, segmentation can be realised in many ways, and the basis for separation of two regions can take many forms. This chapter very briefly describes some aspects of image segmentation, and a way of determining the quality of partitions.

7.1 Where to draw the line - the concept of objects

Segmentation is of course, by its very nature task dependent. The notion of "objects" we ourselves have varies from situation to situation. Consider for instance, a *house* composed of *bricks*. This is something most of us would regard as a *house* first, and a collection of *bricks* second, because we have learned the (to us) meaningful hierarchical notion that one is made up of the other, and that taking individual *bricks* into account makes sense when building a *house*, but not necessarily otherwise. More ambiguous cases arise when taking affordances (the possible actions an object may be involved in, see (Gibson, 1977)) into account. Imagine partitioning a scene into to natural and man-made objects, into graspable and non-graspable objects, or into humans and non-humans. These results will obviously be different, even though the underlying visual information is the same. An example of this kind of ambiguous situation is illustrated in figure 7.1. Thus, there is neither a strict definition of what an object category is, nor is a semantically meaningful model of the world composed of a simple hierarchy of objects. Nevertheless, segmentation of objects is important for recognition since it distin-

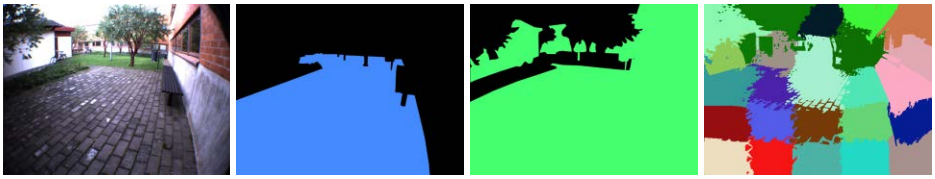

Figure 7.1: *Examples of image segmentation to illustrate the ambiguity of the segmentation task. From left to right: example image, manual segmentation into ground plane and non-ground objects, manual segmentation into man-made and natural objects, superpixel segmentation using simple linear iterative clustering (SLIC) (Achanta et al., 2010).*

guishes information pertaining to the object from that which depends only on background or context.

So what cues are important for separating "objects" from their surroundings, and from each other? Features such as colour, texture and three-dimensional structure are all common candidate features for segmentation. Prior knowledge or assumptions about shape and scale are likely also of importance.

Methods for segmentation vary greatly in their implementation, but almost invariably treat segmentation as a clustering problem on a feature space composed of a combination of spatial and feature dimensions. This can then be solved by, for instance, a greedy approach such as the incremental calculation of *watersheds* (Beucher and Lantuéjoul, 1979), or energy minimisation techniques such as *graph cuts* (Greig et al., 1989) or *superparamagnetic clustering* (Blatt et al., 1996).

7.2 Segmentation and image representation

When it comes to images, and functions thereof (such as measures of local structure, orientation or depth information), the issue of representing these in a way suitable for meaningful segmentation is important to consider. When working with several very different sources of information, computed from visual input in various ways, it is not straightforward to define the "proper" combination of these for the task at hand. Different measurements lie in different ranges, have different noise characteristics, and may or may not be possible to compute at all locations. One way of combining these in a way that is convenient for comparison, is through the use of a *channel representation* (Granlund, 2000). In such a representation, the values are represented by their projection onto a set of basis functions, and the resulting *channel vectors* then represent points in a high-dimensional space including all the features. An important advantage of this is that distances calculated on these vectors behave as sigmoid functions on the original feature spaces, and thus provide a robust error measure. This can be used to obtain better generalisation in segmentation compared to other techniques, and is the subject of (Wallenberg et al., 2011a) (see part II, paper C).

7.3 What is good segmentation?

Since the object concept is ambiguous, so is the idea of a "good" segmentation. In this case, it is necessary to define "goodness" in terms of the expected utility of a segmentation output. The most straightforward way to accomplish this is to specify the desired output and through supervision provide feedback to the system during optimisation of the parameters and representation used for segmentation. While teaching by example might seem like taking the "easy way out", rather than trying to explicitly model the segmentation criteria, it is a natural choice of method in a learning system, where the demands on the solution may change depending on the situation. When segmentation corresponds to division into object classes, one obtains a combined classification and segmentation problem where both object class and spatial extent are to be inferred. This approach is typically referred to as *semantic segmentation*, and is currently a very active research area (for some recent examples, see for instance (Dai et al., 2015) and (Shelhamer et al., 2016)).

7.3.1 Performance measures

In the supervised case, measuring the quality of an image partitioning produced by a segmentation algorithm amounts to somehow describing its similarity to the desired result. This should be done in such a way as to provide a meaningful cost function for further optimisation. The performance measures used in (Wallenberg et al., 2011a) (see part II, paper C) are an attempt to construct measures similar to the *precision* and *recall* used in binary classification problems that are applicable to the multi-way segmentation problem with unknown region correspondences. The basic principles underlying the design of the resulting *consensus score* are *region coverage* and *region specificity*. The motivation for *region coverage* is the assumption that each ground-truth region should overlap a region in the segmentation output, and that each output region should also overlap with a ground-truth region. The higher the overlap, the stronger the connection between the two regions. The trivial solution to this is of course that either the ground-truth or the segmentation result consists of a single all-encompassing region, which is most likely not the desired result. Therefore, the notion of *region specificity* is incorporated by first assuming correspondence between regions with the highest, and then penalising overlap with other regions. In this way, the *region coverage* measure penalises oversegmentation, and the *region specificity* penalises undersegmentation. The final consensus score is the composed of a normalised average of such terms. For further details, see (Wallenberg et al., 2011a) (part II, paper C).

8

Description, Learning and Representation

In order to learn and recognise objects from image data, methods for extraction of object traits, learning of object models, and finally matching and classification based on these must be investigated. This chapter serves to provide an introduction and overview of some such techniques. For further details of methods directly related to this thesis, see chapters 9, 10 and 11.

8.1 What is image content anyway?

The question of the nature of visual information is both inevitable and, very probably, unanswerable. Since the way biological vision systems abstract from visual input is not well understood, the question of representation is still a very open one. Within computer vision, it is long-standing tradition to regard object recognition as a two-stage feed-forward process, in which an image is first converted into a semi-abstract feature representation, and then compared to previous exemplars for retrieval or categorisation purposes (see for instance (Felzenszwalb et al., 2010) and (Sivic and Zisserman, 2003)). Using digital imaging, impressions of the real world are summarised by a combination of responses to a few selected electromagnetic frequencies. These responses then encode the visual information about the scene being viewed. A similar mechanism within the eyes of living creatures sends functions of selected frequency responses to the nervous systems of their owners for interpretation (Land and Nilsson, 2002). Thus, the wealth of visual information we read into what we see *can* be derived (at least in part) from these kinds of observations. Much of the information is there, but the questions of extraction, abstraction and interpretation remain.

8.1.1 The descriptiveness-invariance trade-off

Biological vision systems display a remarkable invariance to many real-world effects. Changes in pose, illumination, shape and the presence of distractors often pose little to no challenge for these systems when recognising objects. There is evidence (see for instance (Perrett and Oram, 1993)) that this invariance is gradually learned, and that a particular previously seen object can only be recognised within a fairly limited appearance range. However, these systems in their complete form are largely invariant to many changes, and can handle significant variations without becoming confused about object identity. Inherent in this is the descriptiveness-invariance trade-off. For a given model complexity, the descriptive power of the model decreases with its invariance to changes in input. In general, it is true that a finely tuned and very specific model has limited applicability outside its own small "comfort zone". In the human visual system, with its massively parallel, complex, and not-quite-hierarchical processing structure, there is ample space for incredibly descriptive object models, capable of representing a huge range of variation. However, this comes at the cost of billions of neurons required to learn and encode this in the very structure of the brain. In the early years of computer vision (concerned primarily with visual pattern recognition), this kind of "storage and processing architecture", usually in the form of an artificial neural network, was common (see for instance (Riesenhuber and Poggio, 1999)). However, as focus shifted from the purely academic toward algorithms for applications outside the research community, the costly, slow-trained (and only implicitly known) features described by these early solutions became less desirable. Much work had been done on the invariant detection and description of certain types of image structures, such as corners, lines and other shapes using *keypoint detectors* and *feature descriptors*. These kinds of representations could, although they described only a small part of the image data, be very robust to many image transformations. They could also be combined to create more complex representations. With this approach however, the descriptiveness-invariance trade-off on the level of individual image regions is fixed, such as in a Bag-of-Words or part-based model (see 8.1.3) and the flexibility lies in how groups of these are matched. In recent years, likely due to the increase in computing power available in many devices and the focus on parallelism in image processing, methods have tended to rely less on these hand-crafted semi-abstract descriptions of image content. Using techniques based on universal approximators (primarily decision forests (Özuysal et al., 2007; Kalal et al., 2012) and, after the success of Krizhevsky et al. (Krizhevsky et al., 2012), neural networks) end-to-end learning is now common once again.

8.1.2 Dense versus sparse representations

The digital images encountered in computer vision can probably most easily be thought of as a discrete approximation of an underlying visual world (whose continuous or discrete nature it seems the physicists are having a hard time deciding on). In their simplest form, they are made up of pixels - the contents of spatially

arranged "photon bins" (in the case of colour imaging, a Bayer filter[1] is the most common arrangement), accumulated over a specific *integration time*. A grid of pixels is then generated from these values. In this sense the images themselves constitute a sparse sampling (in both space, time and frequency) of the visual world. However, when regarding image descriptions, the convention is to regard methods that generate values at (at least) all pixel locations as dense, and those that generate values at only a small subset of locations as sparse. In many cases the sparse representations are of higher complexity, and restricting the number of computed descriptors is necessary performance reasons. The sparse representations instead often rely on the notion of *keypoints* that can be reliably detected and used to describe a subset of important image features.

8.1.3 Ordered and unordered representations

Regardless of whether a dense or sparse representation is used, the question of how to handle the spatial structure of the world is an important one. Spatial information and context are clearly important for object recognition in biological systems, and our notion of "objects" is often related to spatial entities. However, complex structures are notoriously difficult to describe in a way that is invariant to real-world changes (in for instance pose, illumination, occlusion, etc.). Local structures (usually termed *keypoints*) can often be robustly described due to their limited variability under geometric transformations. Some methods (like the Bag-of-Words method, see section 8.3.1) leave the invariance at the level of the individual keypoint and simply ignore spatial arrangement. While this results in invariance to permutation of keypoints and some level of robustness to partial occlusion, it inevitably sacrifices descriptive power. Other unordered representations include features based on global histograms, or other properties not encoding spatial relationships between measurements. An intermediate class of representations retains some coarse-level spatial structure, an example of this is the class of models termed *part-based models*[2], which consists of a constellation of deformable parts capable of moving relative to each other. The most rigidly structured model is of course the image itself, possibly represented in a canonical coordinate frame and used for *template matching*. Illustrations of the structure of these representations are shown in figure 8.1.

8.2 Commonly used descriptors

It seems that in most cases, biological vision at its most basic level is based more on contrast than intensity (Land and Nilsson, 2002). It is therefore not surprising that many of the image descriptors that explicitly aim at invariance are based on local contrast in the form of image gradients. The most common example

[1] The Bayer filter is the most common *colour imaging array*, and was developed at Eastman Kodak. For details, see US patent 3971065A.

[2] Earlier publications use the term *pictorial structures* (Fischler and Elschlager, 1973). A more recent example is the *deformable parts model* (Felzenszwalb et al., 2010).

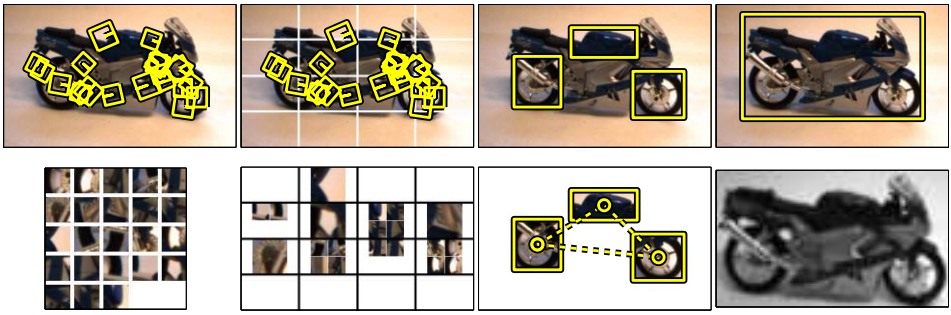

Figure 8.1: *Illustration of image representations in order of increasing structural rigidity. Top row shows the image regions used, bottom row illustrates the structure of the representation. From left to right: global Bag-of-Words, one level of a spatial pyramid Bag-of-Words, part-based model, template.*

of this is the immensely popular *scale-invariant feature transform* (SIFT) (Lowe, 2004), which combines keypoint detection and scale selection with a local orientation estimate and gradient histogram. For a corner-type structure, it thus generates estimates of location, scale, orientation and local gradient distribution in a scale- and orientation-normalised coordinate frame. Other examples of common descriptors based on similar techniques include *gradient location and orientation histograms* (GLOH) (Mikolajczyk and Schmid, 2005), *speeded-up robust features* (SURF) (Bay et al., 2006) and *histograms of oriented gradients* (HOG) (Dalal and Triggs, 2005), all of which have been applied to matching and recognition tasks. Recently, binary descriptors such as *binary robust scalable invariant keypoints* (BRISK) (Leutenegger et al., 2011) and later *fast retina keypoints* (FREAK) (Alahi et al., 2012) have gained popularity due to the speed with which they can be calculated (especially when combined with a fast keypoint detector, such as the *features from accelerated segment test* (FAST) (Rosten et al., 2010) corner detector). At the other end of the spectrum, pre-trained neural networks for large-scale image classification tasks have also been found to generate remarkably useful feature sets for a multitude of problems. For an overview of such applications, see for instance (Razavian et al., 2014).

8.3 Learning and inference

In the context of computer vision, the description of image content is almost always done for one of two purposes, geometrical matching or image-based attribute estimation/retrieval. The aim of a descriptor is thus to reflect image content in such a way that it can be reliably and robustly matched to other images, either for establishing a relationship between images or in order to learn an abstract description based on multiple observations. While reconstruction and geometry are mainly concerned with the former, object recognition deals mainly

with the latter. The goal of description is thus to create a persistent representation that, based on observed object instances, encodes object information in a way that is suitable for comparison to novel observations. The purpose of this persistent structure (the object model), is then to make inferences about these new observations, and to test hypotheses regarding attributes of the object. Usually, this involves training one or several classifiers to discriminate between models. These classifiers can be of varying complexity, and express their decision boundaries in various ways. Common examples of classifiers include *k-nearest neighbour* (k-NN) classifiers, *neural networks*, *decision trees*, and *support vector machines* (SVM) (Cortes and Vapnik, 1995)[3]. These classifiers then (individually or in combination) provide an object hypothesis and (possibly) an indication of the degree of confidence in this hypothesis.

8.3.1 Bag-of-Words methods

Bag-of-Words methods were an important step in computer vision due to their ability to very robustly and compactly describe images in a way suitable for whole-image classification and matching. Initially, these were typically based on local invariant features, but the technique is in general separate from any particular choice of local descriptor. Further details about these methods, and their relation to work presented in this thesis can be found in chapter 9.

8.3.2 Single-classifiers and ensemble methods

In many applications it is very difficult to achieve good classification results due to the limitations of a particular classifier. One approach to handling this is increasing the complexity of the classifier to accommodate for complex decision boundaries, but this also means that the training of this classifier becomes more sensitive, time-consuming and in general requires more training data. Another option is to train multiple classifiers based on either different sets of training data or different features, and then combine their outputs in a probabilistic fashion. Since this procedure is based on the notion of statistical ensembles, these are termed *ensemble methods*. Ensemble methods are often used in decision tree learning, the most common example being *random forests* (Ho, 1995; Breiman, 2001) and related methods. Despite their heavy memory requirements, these techniques allow many (typically thousands) of very weak learners to jointly describe complex high-dimensional decision boundaries and achieve classification performance they would individually be quite incapable of. Some methods of this class are also well suited to parallel execution and incremental training, making them applicable to systems that require online learning functionality. A more in-depth description of decision tree ensembles can be found in chapter 10. Handling high-dimensional data using such ensembles is also discussed in (Wallenberg and Forssén, 2016) (see part II, paper E).

[3] For an overview of these "classical" techniques, as well as more modern ones, see for instance (Hastie et al., 2013).

8.3.3 Neural networks and deep learning

Neural networks are among the most basic of learning models, in that their power (like that of the decision forests) lies in the combination of many weak learners (neurons). However, due to the construction not being as straight-forward or easy to analyse as purely discriminative methods and the computational and numerical challenges involved in training such models, it is only recently that they have seen major mainstream popularity. However, with the rise of generally available parallel processing hardware for images, large-scale neural networks are now among the most promising avenues of machine learning and computer vision research. An overview of this class of methods as it relates to the contents of this thesis is given in chapter 11. A comparison of mechanisms for incorporation of visual attention into deep networks is also the subject of (Wallenberg and Forssén, 2017) (see part II, paper F).

9

Bag-of-words methods

This chapter gives a brief introduction to computer vision applications of *Bag-of-Words* (BoW) methods. These methods were an important step towards large-scale image retrieval and robust image matching using local invariant features. While the primary choice of image features may have changed in recent years, bag-of-words representations are still used as robust and compact input data for large-scale machine learning problems.

9.1 The Bag-of-Words method in computer vision

The *Bag-of-Words* model, in vision applications also referred to as *Bag-of-Visual-Words* or *Bag-of-Features* (Sivic and Zisserman, 2003), has its roots in document retrieval and relies on comparing two texts using a "vocabulary" of available words and a histogram comparison. Each text is parsed, and occurrences of words recorded in a histogram. The frequencies of these words then constitute the description of the document. Since this representation does not encode any ordering of the words, any permutation of the same sequence yields the same representation (hence the moniker). This of course also means that the matching is invariant to any permutation of the document. Since the size of the vocabulary determines the dimensionality of the description, it also determines the computational cost of matching. A larger vocabulary allows for more specific search terms (i.e. higher descriptiveness), while it is more sensitive to missing or distorted information (i.e. less invariant). The application of this kind of matching to images comes from the need for an efficient image retrieval system for large-scale image search (Sivic and Zisserman, 2003; Csurka et al., 2004). In the image case, the vocabulary consists of a set of prototype descriptors (commonly referred to as *visual words*), which are used as a "code book" to convert an image into a histogram of visual word occurrences. The visual word histogram retains both the

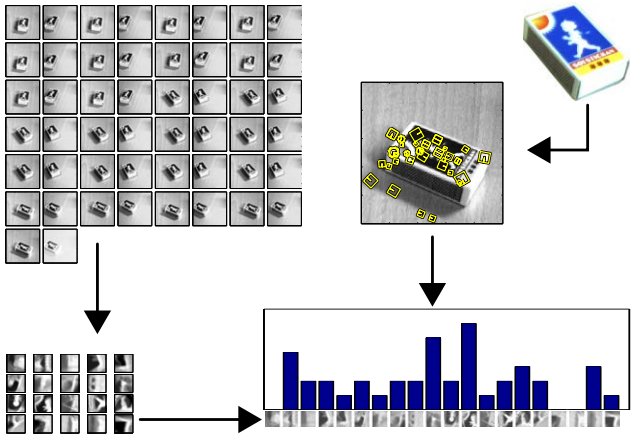

Figure 9.1: *Illustration of the BoW model using a small set of 20 prototype features. Prototype features extracted from a large number of images (left) allow an object (the matchbox) to be represented using a histogram of these prototype features extracted from an image of the object (right).*

permutation invariance of the original BoW histogram, as well as any invariances provided by the descriptors themselves (such as scale and rotation invariance) at the cost of descriptor quantisation and loss of spatial ordering. An illustration of a BoW model can be found in figure 9.1.

9.1.1 Vocabulary generation

Since the vocabulary used to describe query images has to represent all the image data the system will ever encounter, its descriptive power is important. The vocabulary must also be of a tractable size (although the definition of this depends on the constraints imposed by the intended applications). The most common approach, as used in (Sivic and Zisserman, 2003) (and also in (Wallenberg and Forssén, 2010b,a), see part II, papers, A and B) is to sample a large set of image features (in these cases described by SIFT descriptors (Lowe, 1999, 2004)) and then cluster these to obtain a smaller set of prototypical features, the assumption being that this clustering will capture the feature-space structure of the images. Typically, the number of vocabulary items can vary from a few hundred (in cases where speed is key) to several million in large-scale image retrieval tasks, where specificity must be retained and speed is less critical than accuracy.

9.1.2 Learning

Once the vocabulary is generated, training samples can be created from images by computing keypoint descriptors and quantising them using the vocabulary.

Learning, in the simplest case (as in (Wallenberg and Forssén, 2010b) and (Wallenberg and Forssén, 2010a), see part II, papers A and B) consists of accumulating and storing the visual word histograms of multiple observations of an object. Since image features are typically not evenly distributed across the possible feature dimensions, weights can also be calculated to increase the contribution of rare (and therefore more discriminative) features. An example of such a weighting scheme is the *term frequency-inverse document frequency* (TF-IDF) weighting scheme (Jones, 1972; Salton and Buckley, 1988). This kind of weighting scheme serves to compensate for varying term (feature) density among exemplars, and to emphasize features suitable for class discrimination.

9.2 Confidence and hesitation

When presented with an object hypothesis generated by a classifier, one typically also wants a measure of how stable this hypothesis is, and how much faith to have in it. This brings up the notion of *confidence*, and its rate of change over time (which will be referred to as *hesitation*), and how these are related to the matching procedure used. Comparison of BoW histograms can be accomplished in any of several ways, either through use of standard distance measures such as L^p-norms, or by more sophisticated distance measures such as the *earth mover's distance*[1]. Another common similarity measure for both histograms of populations and vectors in general is the *cosine similarity*[2] measure, derived from the fact that for vectors in Euclidean space, the inner product of two vectors is equal to the product of their respective magnitudes and the cosine of the angle between them. If the weighted BoW histograms are normalised to unit length, this means that their inner product is exactly equal to the cosine of this angle (and the angle itself describes the geodesic on the unit sphere in the space of the weighted vocabulary). Since the dimensionality of these vectors can be high, and since efficient tools exist for matrix-vector operations, this is a simple and attractive choice of similarity measure. Confidence and hesitation measures based on this kind of similarity are discussed in (Wallenberg and Forssén, 2010b) and (Wallenberg and Forssén, 2010a) (see part II, papers A and B).

9.2.1 Confidence measures

Since multi-way classification is essentially based only on discrimination *between* classes, the definition of confidence is not straightforward. A matching procedure can only give information about the similarity (or dissimilarity) of an observation to a number of other observations, and the distribution of these values must then be used to determine whether or not a conclusive decision can be reached. In general, given the similarity or dissimilarity of a query to a set of previously encountered prototypes in a multi-way classfication setting, an ab-

[1] In mathematics, this is usually referred to as the *Wasserstein metric*.

[2] It is also known by other names, such as the *Jaccard index* (Jaccard, 1901) or the *Tanimoto coefficient* (Tanimoto, 1957).

solute similarity/dissimilarity (if it lies within a known range) says something about the confidence that an observation matches a known exemplar, but nothing of the specificity of the match. On the other hand, a purely relative similarity/dissimilarity measure can describe how likely one match is compared to another, but lacks absolute scale. This implies that the former can be of use when determining whether or not any match can be established (or determine whether the query is *known* or *unknown*), and that the latter is suitable for determining the class confidence once this distinction has been made.

9.2.2 Confidence gain and hesitation

"Look up in the sky... It's a bird... It's a plane... It's Superman!" might be the output of a very confused classifier, teetering on the decision boundary between multiple object classes. Clearly, this behaviour shows not only that the matching procedure is sensitive, but also that, given time, the correct hypothesis can emerge as the number of observations (and thereby the amount of cumulative information) increases. The question of when to make a decision about an observation is an important one, and also one which is in principle impossible to answer since one cannot know what the next observation will bring. This raises the issue of predicting viewpoint utility from a small number of observations. If time allows, and it seems that a new observation will bring significant additional confidence to the decision, it may be advantageous to postpone output until further observations have been made. If, on the other hand, there is no time for another observation because the decision has to be made or if it seems additional observations cannot help resolve ambiguities, the best course of action is probably to go with the currently best hypothesis. An investigation of this is the subject of (Wallenberg and Forssén, 2010a) (see part II, paper B).

10

Decision forests

This chapter gives a brief introduction to decision forests in the context of work presented in this thesis. The basic structure of decision trees, decision forests and aspects of feature and discriminant selection proposed in (Wallenberg and Forssén, 2016) are introduced. For a more detailed description of decision tree learning and its applications, see for instance (Hastie et al., 2013).

10.1 Decision trees and decision forests

In this section, the concepts of decision trees and decision forests will be introduced. The basic structure of these estimators will be illustrated, and examples of fusion methods will also be given.

10.1.1 Decision trees

Decision trees have a long history within computer science and machine learning. The decision tree is a branching structure consisting of internal nodes and leaf nodes. The internal nodes are also referred to as *weak learners* due to their individually limited discriminative capabilities. At each such internal node, a test resulting in a data split is performed, and at each leaf node some information about an attribute or conclusion drawn from these tests is stored. An example of a small binary decision tree is shown in figure 10.1.

Typically, binary trees are used due to their ease of construction and speed and due to the relatively simple statistics of binary classification problems. In this case, each internal node of the decision tree partitions the data space into two parts, and through the hierarchical application of many such splits, the data space is partitioned into decision regions separated by sets of split functions. In the simplest case, the decision boundary is based on a single data dimension, and

the split is determined by a threshold value along that dimension. However, any operation resulting in a scalar function to which a threshold can be applied could be used. Common choices include linear or affine functions, corresponding to decision boundaries made up of sets of hyperplanes in the data space. Due to the simplicity of each such operation, and also the fact that only a subset of nodes need to be visited when traversing such a tree, these estimators are among the fastest available. An interesting special case is the *fern* structure, where all internal nodes at a particular depth contains the same split function. This allows for parallel application of all splits, resulting in a very fast and lightweight estimator (see for instance (Özuysal et al., 2007)) at the cost of reduced discriminative power.

Once the split functions within a tree branch have been applied, a leaf node is reached. The model or predictor stored in this leaf node determines the type of estimator the tree represents. It could, for instance, be frequencies of sample classes within the decision region or the parameters of some other attribute distribution (age, sex, colour, location). This means that the decision tree can be used for either classification or regression by selecting an appropriate leaf node model.

If the depth of the tree (and thus the number of internal nodes) is not limited, the decision tree is a *universal approximator* that can express any function on the data space by sufficiently local piece-wise approximation. In practice, however, there is a practical limit to the tree depth due to sample density, computational limitations or measurement noise. If measurement noise is present, any partitioning will become brittle with sufficient subdivision and the estimator will not be robust or generalise well to novel data[1].

10.1.2 Decision forests

Although the decision tree can potentially describe any decision boundary at sufficient depth, deep trees have some inherent problems (as described above). When dealing with high-dimensional, noisy or diverse data, a more robust and tractable solution is to use not one, but many decision trees. These tree are combined in an ensemble, commonly referred to as a decision forest. For practical details about the implementation of decision forests, and their applications to a wide range of estimation problems, see for instance (Criminisi et al., 2012). Arguably, one of the most successful decision forest methods is *random forests* (RF) (Breiman, 2001). A particularly successful large-scale application is the player pose estimation used on the Microsoft Kinect (see (Shotton et al., 2011)).

What all these techniques have in common is that they seek to combine not only many individually weak but complementary discriminants, but also many weak but complementary trees. This is achieved by training trees on data subsets, potentially allowing each tree to solve a simpler (ideally uncorrelated) problem and then combining these to solve the more complex problem at hand. If an ap-

[1] Breiman and Cutler argue on their Random Forests web-page that this does not constitute overfitting (see `https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`). The author, however, disagrees.

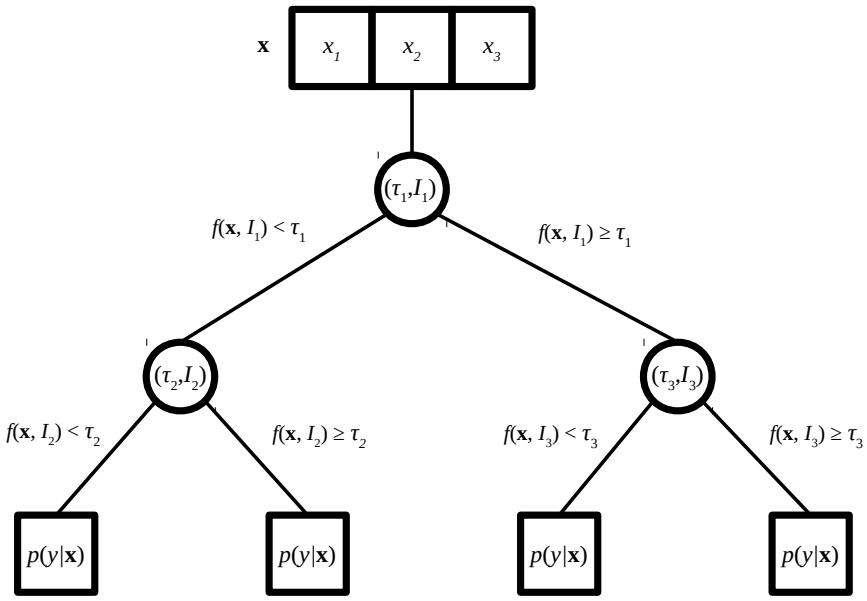

Figure 10.1: *Example of a small decision tree. The three-element input vector* $\mathbf{x}$ *is mapped to a distribution* $p(y|\mathbf{x})$ *on the output* y*. Each internal node, indicated by a circle, contains a threshold value* τ_j *and a subset of input dimensions* I_j*. The function* f *is applied to the subset* I_j *and the result is then compared to the threshold* τ_j*. Depending on the outcome, either the left or right branch is chosen at each node. The leaf nodes contain the resulting output distribution* $p(y|\mathbf{x})$ *indicated by the node test results along the path to the leaf.*

propriate method is used to fuse the tree outputs, this can provide much greater robustness to noise and outliers, even when a common feature space cannot be found.

Fusion approaches include voting among trees (especially in classification scenarios) and a variety of mode-finding approaches such as finding the average or median or using kernel density or mean shift methods. Which alternative is best suited depends on the nature of, and the requirements on, the desired output. At run-time, fusion can also be applied to input or output data (for instance, across multiple frames in a video, multiple images in a sequence or multiple time windows of an audio signal). This is usually termed *early fusion* when done in the input data space and *late fusion* when done in the output space.

10.2 Feature selection and learning

In this section aspects relating to the training of decision forests will be described. Specifically, the concepts of data distribution by bagging, feature selection using canonical correlation analysis and an efficient method for selection of sparse hyperplane discriminants will be described.

10.2.1 Bagging

Bagging (or *bootstrap aggregation*) is the term applied to the partitioning of data across trees in decision forest methods. A good bagging method partitions data such that each tree receives a subset of data that is simpler to partition than the complete dataset, and that reflects a different aspect of the data than that received by the other trees. This can be done by assigning each tree different subsets of samples, different subsets of data dimensions, or a combination of both. The selection can be either random (as in (Breiman, 2001)) or guided by prior knowledge of the measurements, such as that they correspond to specific image locations or specific modes in the data space.

10.3 CCA-based feature selection

While decision forest estimators can partition a data space into arbitrarily small decision regions, the top-down hierarchical partitioning means that neighbourhood relations between samples are preserved within each decision region. The underlying assumption is that the input space (at least locally) has a semantically meaningful distance corresponding to the attributes to be estimated. However, if this is true only locally, this may result in many tree levels being used only in order to obtain these decision regions, many similar local solutions at several places within the same tree and an estimator which is computationally costly and sensitive to noise. In these cases, a more appropriate feature space should be found.

Dimensionality reduction techniques based on variance preservation such as *principal component analysis* (PCA) (Pearson, 1901) do not address this problem, since the large-scale variations in data are not the carriers of semantically important information. A better alternative within the same class of methods is *canonical correlation analysis* (CCA, see for instance (Borga, 2001)). The advantage of CCA is that, instead of finding eigenvectors that preserve data variance, it finds a joint maximising correlation of two sets of data. This means that a feature space with a semantically meaningful distance can be found by applying CCA to the input data and the attributes to be estimated. This can be done for several attributes at once, creating a common feature space useful for estimation of several attributes. A procedure for finding such spaces is described in (Wallenberg and Forssén, 2016) (see part II, paper E). Another recent method that addresses this problem using neural networks (a more computationally costly, but also more flexible approach) is used to train the FaceNet network (Schroff et al., 2015).

10.3.1 Discriminant selection

Training of decision forest estimators consists of two parts: selection of decision boundaries in the internal nodes and fitting/storing the models in the leaf nodes. The method used for selection of decision boundaries determines has the largest influence on the time taken to train the estimator, and on its final performance. Alternatives range from fully randomised selection of single dimensions or pairs thereof (see (Özuysal et al., 2007; Kalal et al., 2012)) to randomised hyperplanes or rotations (see (Breiman, 2001) and (Rodriguez et al., 2006), respectively). The process of sampling these discriminants determines both the training time and the randomness of the final estimator. If discriminants are sampled completely at random, training time consists only of passing the available training data through the decision forest and fitting the leaf node models. However, if tree depth and forest size are limited, the likelihood of finding enough useful splits is low. Therefore, a more common approach is to select a number of candidates at each internal node, to increase the chances of finding a useful split. Each sampled discriminant is typically ranked by its *information gain* (a measure of the change in entropy before and after the split) and sampling continues until either a set value is reached or a set number of samples have been drawn, after which only the best discriminant found is retained. Since each candidate to be evaluated incurs a cost in the form of training time it is desirable that the ratio of potentially useful candidates is as high as possible, and preferably reflects a semantically important aspect of the data. Also, since a sparse set of data dimensions is faster to evaluate than a denser set, a method for selecting sparse sets of highly discriminative data dimensions is also desirable.

This process can be done using minimal sample subsets, in a similar fashion as in the *random sample consensus* (RANSAC) estimation framework (Fischler and Bolles, 1981). Given two samples (a minimal subset for the definition of a hyperplane) a maximally separating hyperplane discriminant can be calculated. If these are selected such that they are semantically different (for instance, if they belong to different classes) a discriminant based on the resulting separating hyperplane is likely to be a useful indicator of this difference. Then, provided the feature space distances are semantically relevant, this can be greedily sparsified while retaining as much discriminative power as possible for a set maximum number of feature space dimensions. A method for selecting discriminants in this way is described in (Wallenberg and Forssén, 2016) (see part II, paper E).

11

Neural networks and deep learning

The resurgence of neural networks may be one of the most important steps toward a capable and general framework for object recognition in recent years. This chapter aims to give a brief introduction to selected aspects of neural networks and deep learning in the context of this thesis. Specifically, the aim is to provide context for the bottom-up attention mechanisms investigated in (Wallenberg and Forssén, 2017) (see part II, paper F). For a more comprehensive summary of modern neural network techniques, see for instance (Goodfellow et al., 2016).

11.1 Artificial neural networks

The concepts and ideas behind *artificial neural networks* (ANN:s) sprang from research on computational models in neuroscience. Early work on mechanisms for neural computation (see for instance (McCulloch and Pitts, 1943) and (Hebb, 1949)) laid the foundation for the *perceptron* architecture (Rosenblatt, 1958), which became the basis for much of the later work on neural networks. Inspired by studies of the mammalian visual system (Hubel and Wiesel, 1959), one major application of interest was pattern recognition. There was, however, a lack of efficient training methods for these networks. This was addressed with the introduction of the *error back-propagation* (EBP) method (Werbos, 1974), which allowed supervised training of any network, provided error gradients could be calculated for each layer. The concept of *receptive fields* in the visual cortex was also an important component of the (then) highly successful Neocognitron network (see (Fukushima, 1988) for what may be the definitive version) for translation- and rotation-invariant pattern recognition. A simple example of a small fully-connected ANN structure can be found in figure 11.1. Further details about its components can be found in section 11.1.2.

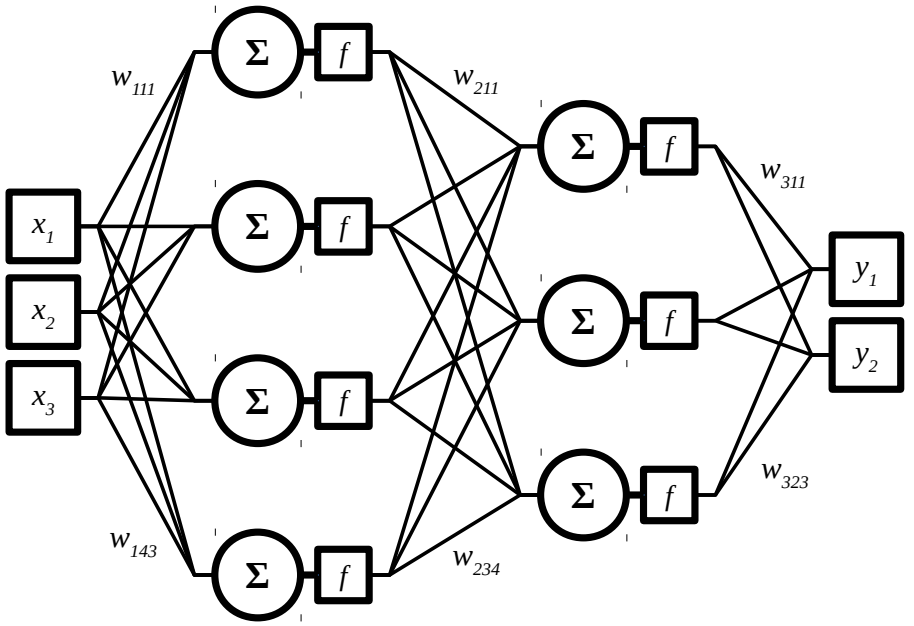

Figure 11.1: *An example of a small fully-connected artificial neural network. The inputs x_n are mapped to outputs y_m via the intermediate neuron layers, indicated by circles. Each node receives inputs from every output on the previous level. Each incoming connection i at a node j in layer k has an associated weight w_{kji} which is multiplied onto the incoming value. These are then summed within the node and passed to an activation function f. The resulting output is then passed as input to the next layer.*

11.1.1 Convolutional neural networks

While the Neocognitron contained elements similar to receptive fields, these were applied at fixed positions within an input. Although this meant that input elements were shared between receptive fields and that these *cells* (as they were called) were therefore locally similar, a large number were still required.

Many signals (such as sound, images and video) contain semantically important local neighbourhood structure (spatially, temporally or both). For instance, a word is temporally localised within an audio signal and an object is spatially localised within an image. The word and the object do not in themselves change depending on where within the signal they are found (although their interpretation and importance may depend on the context). For fixed-resolution signals, this means that it is reasonable to assume translation invariance and that the same pattern should generate the same output, regardless of its position within

the signal. In practice this means that weights in a neural network can be shared across receptive fields, and that the entire signal can be processed using convolution or correlation with a kernel representing the weights within a receptive field[1]. One of the early successes of such a *convolutional neural network* (CNN) was the LeNet (LeCun et al., 1998), which combined multiple layers of such filter banks. Introducing filter banks into neural networks resulted in a massive reduction in the number of weights required to process large signals. The sharing of weights also meant that every signal position could be used as training data, giving these networks a better chance of converging to a meaningful solution during training. However, as convolutional operations are costly and time-consuming on hardware not designed for parallel processing, and as such hardware was rare at the time, these systems were not widely used in practice. An example of a small CNN can be seen in figure 11.2. Further details of its components can be found in section 11.1.2.

11.1.2 Basic operations and structure

Although the structure and makeup of artificial neural networks varies greatly, there are common elements that are used in most networks. These will be briefly described here.

Fully-connected layers

In a fully-connected layer, each neuron in the network received input from all neurons in the previous layer. These layers were the main component of early neural networks. Due to the all-to-all connectivity the number of weights required in such a layer grows quadratically with the number of neurons, making fully-connected layers computationally expensive for high-dimensional data. Training such layers requires a large amount of training data compared to other optimisable layer types, and also requires the input to be of a fixed size.

Convolutional layers

In a convolutional layer, the input is convolved or correlated with one or more convolution or correlation kernels. This can be done in a spatially dense or sparse fashion, the distance between applications of the kernel often being called the *stride* of the convolutional layer. The number of weights required depends on the spatial extent of the kernel and the number of data dimensions the kernel is applied over. The operation of the convolutional layer is translation-invariant and does not depend on a fixed input size.

[1] Many implementations prefer correlation over convolution due to its simplicity for real-valued signals. However, they do not hesitate to refer to the resulting operations as "convolutions", which sometimes leads to confusion.

Activation layers

The activation layer applies an element-wise activation or transfer function to each of its inputs. As both the fully-connected and convolutional layers represent linear operations, they cannot represent non-linear functions or solve problem that require non-linear mappings. For this reason, activation functions are typically nonlinear. Common activation functions include *sigmoids* and *rectifiers*. The first of these categories represents *saturating* activation functions, whose co-domains are bounded. The second category is an example of a class of *non-saturating* activation functions lacking an upper bound. Each has its advantages and drawbacks. For further details, see for instance (Goodfellow et al., 2016).

Pooling layers

Spatial pooling layers combine several inputs within a local neighbourhood to produce a pooled output. The most common types of pooling operations are average pooling (where the output is the average of the inputs) and maximum pooling (where the output is the maximum of the inputs). These operations also correspond to those used to combine inputs in (Riesenhuber and Poggio, 1999). Pooling can be applied in dense or sparse fashion, and can be used to propagate strong activations within the network.

Normalisation layers

Normalisation layers (as the name implies) normalise their inputs in some way. Typical normalisations include per-location normalisation across non-spatial dimensions, spatially local normalisation of individual data dimensions and global normalisation of entire signals.

11.1.3 The rise of deep learning

As hardware for parallel processing of signals became more and more common (mainly fuelled by the computer graphics and gaming industries), the interest in artificial neural networks (and CNN:s in particular) grew. A major breakthrough in popularising the combination of multi-layer convolutional networks and "traditional" fully-connected ANN:s was the publication of AlexNet (Krizhevsky et al., 2012), which won the ImageNet Large-scale Visual Recognition Challenge (ILSVRC) in 2012. These networks (now commonly referred to as simply *deep neural networks* or DNN:s) have since come to dominate many areas of signal processing and machine learning.

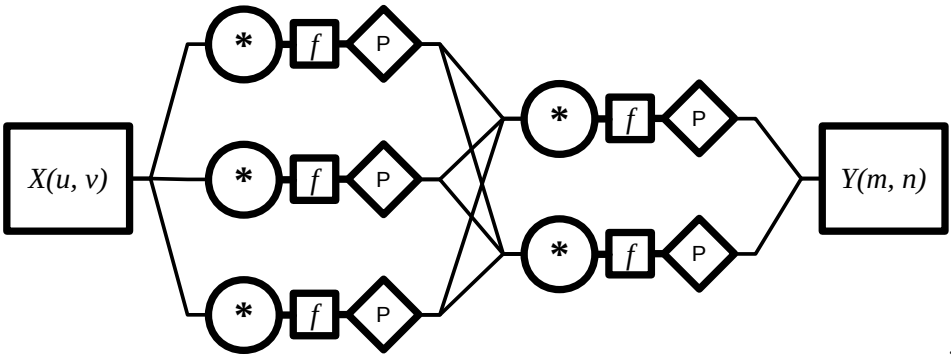

Figure 11.2: *An example of a small convolutional neural network with a 2D array of scalar inputs $X(u, v)$ and a 2D array of 2D outputs $Y(m, n)$. The inputs $X(m, n)$ are mapped to outputs $Y(m, n)$ via the intermediate neuron layers, indicated by circles. Each node receives inputs from every output on the previous level and contains a convolution kernel. The inputs are convolved with this kernel passed to an activation function f. The resulting is then passed to a spatial pooling operator P, after which it serves as input to the next layer.*

Although successful, the early examples of such networks were hard to train, and required (by the standards of the time) huge amounts of training data and computational power. Also, training was not always successful due to the massive number of parameters that needed to be optimised. In recent years, many techniques for improving robustness of the network and convergence during training have been published. Some noteworthy examples include:

- Regularisation by *dropout* (Srivastava et al., 2014), in which a random subset of activations within the network are suppressed during each training iteration. This has the two-fold effect of forcing the network to cope with incomplete data and focussing the particular training iteration on updating a specific subset of weights. When properly applied, this can both improve convergence and result in a more robust final network.

- Data conditioning by *batch normalisation* (Ioffe and Szegedy, 2015), in which a trainable mean-subtraction and variance normalisation is applied to each data dimension. During training, this is useful is both balancing data dimensions and alleviating numerical problems caused by gradients either vanishing or "exploding" when propagated through multiple network layers. This simplifies training parameter selection, allows for more aggressive optimisation and significantly speeds up the training process.

Techniques such as these have brought successful training of deep networks within the grasp of researchers, application developers and hobbyists alike. New network components and architectures have also been introduced, with the aim of improving estimation performance or widening the field of possible applications. Some recent significant innovations include:

- Reintroduction the multi-resolution pyramid for feature aggregation by *spatial pyramid pooling* (He et al., 2014). This addresses the gap between convolutional networks that do not depend on any fixed input size and fully-connected networks with a fixed number of inputs.
- Task-dependent input resampling by *spatial transformer networks* (Jaderberg et al., 2015), which learn to spatially transform inputs such that learning is simplified without depending heavily on user-specified assumptions about the specifics of that transformation.
- Incorporation of known geometry using *group-equivariant convolutions* (Cohen and Welling, 2016), which enforce equivariance of the convolution operations within a CNN under specified operations, such as rotation. This reduces the need to learn invariances to these operations, and also explicitly specifies their effects under ideal conditions.
- Identity mappings and the ResNet architecture (He et al., 2015), which involves bypassing entire computation layers. This is done in order to construct and train very deep and highly redundant networks, which can provide both discriminative power due to their depth and robustness due to those inherent redundancies.

Additionally, there is increasing interest in *recurrent neural networks* that do not use a strictly feed-forward processing architecture. These networks combine feed-forward and feedback connections between layers, much like the higher levels of biological nervous systems. However, since this necessitates taking temporal aspects into account (due to the time required for the network to settle), and due to the challenges inherent in optimising potentially unstable feedback systems, these are not as commonly used.

11.1.4 Pre-trained deep networks for feature extraction

As a result of the many successful DNN-based methods published, such pre-trained networks are readily available for download. Networks pre-trained on large-scale image classification tasks have become widely used as feature extractors for other tasks (see for instance (Donahue et al., 2013), and are also a common starting point for transfer learning between applications. The reason for this is that these networks (particularly in their convolutional layers) capture an efficient encoding of image content. This provides both a good feature set for other estimators and a good starting solution for training other networks. It has even been argued that this type of feature extraction should be the first choice in most applications (see for instance (Razavian et al., 2014)).

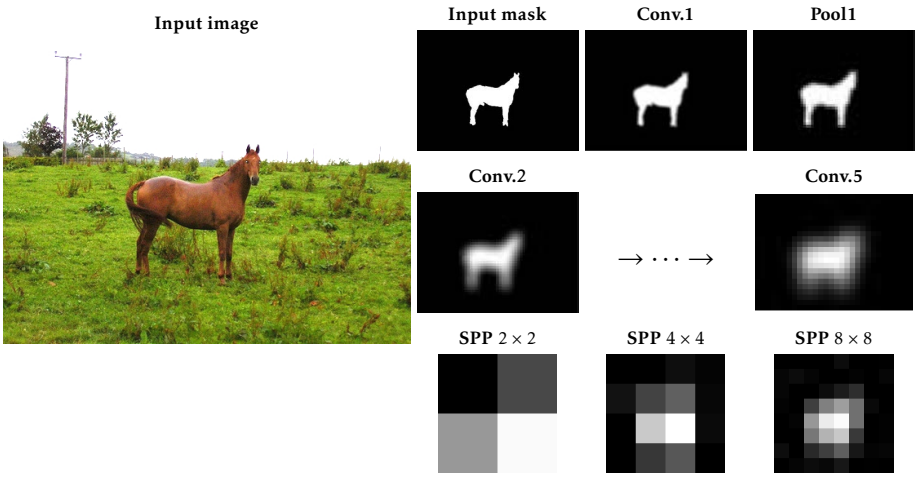

Figure 11.3: *Mask propagation for selected layers within a DNN. Left: input image. Right: Masks propagated through the network. Shown here is the sum over all feature dimensions to indicate the influence of the attended pixels at a particular spatial location. Figure from (Wallenberg and Forssén, 2017).*

11.2 Attention models in deep networks

When processing complex visual scenes, an attention mechanism can serve to disambiguate and isolate specific stimuli (such as objects). This is useful in that it allows a recognition system to "focus" on a particular feature type or region, allowing it to handle cases it would not otherwise be able to solve. In the context of deep neural networks, and within this thesis, the *attention mechanism* refers to operations that are not part of the network structure, but are applied to data at various levels of the network in order to guide processing. For instance, a network trained to find cars may exhibit strong, spatially localised activations when presented with such an object, but this is not an attention mechanism. In contrast, an operation which promotes or suppresses activations based on prior knowledge, independent of the actions of the network *is* (according to this author's definition) an attention mechanism. Likewise, an operation that, given a hypothesis or task, generates a set of candidate inputs (such as an image region) also falls into this category. For instance, given an image of a sheep and a dog, the attention mechanism is that which determines what the desired output is when a specific image location is attended to. This is further complicated by the role of context in recognition, as the context in which objects appear can have a significant effect on how they are perceived (see for instance (Mottaghi et al., 2014)). As stated in chapter 6, computational models may contain both bottom-up and top-down components. However, since top-down attention is closely linked to actions and tasks, it is less straight-forward to study than bottom-up attention.

11.2.1 Bottom-up attention for pre-trained deep networks

Since pre-trained networks are an important asset when the amount of training data or computational resources are not sufficient for end-to-end training, bottom-up attention mechanisms that can work directly with pre-trained networks are of interest. However, these must work in such a way as to allow the pre-trained network to function without disruption, while suppressing non-salient regions. Methods that affect activation statistics, such as the normalisation model (Reynolds and Heeger, 2009) cause problems since they alter the way in which image content is related to activations within the network. Similarly, multiplicative masking techniques can cause artifacts that degrade performance, even when properly applied. Masking methods combining multiplicative suppression with blending have been shown to improve performance when properly applied (see for instance (Walther and Koch, 2006)). An extended multi-layer approach, designed to aid classification despite errors in masking is the subject of (Wallenberg and Forssén, 2017) (see part II, paper F). An example of bottom-up propagation of an attention mask within a DNN is shown in figure 11.3. Effects of various masking techniques investigated in (Wallenberg and Forssén, 2017) are shown in figure 11.4.

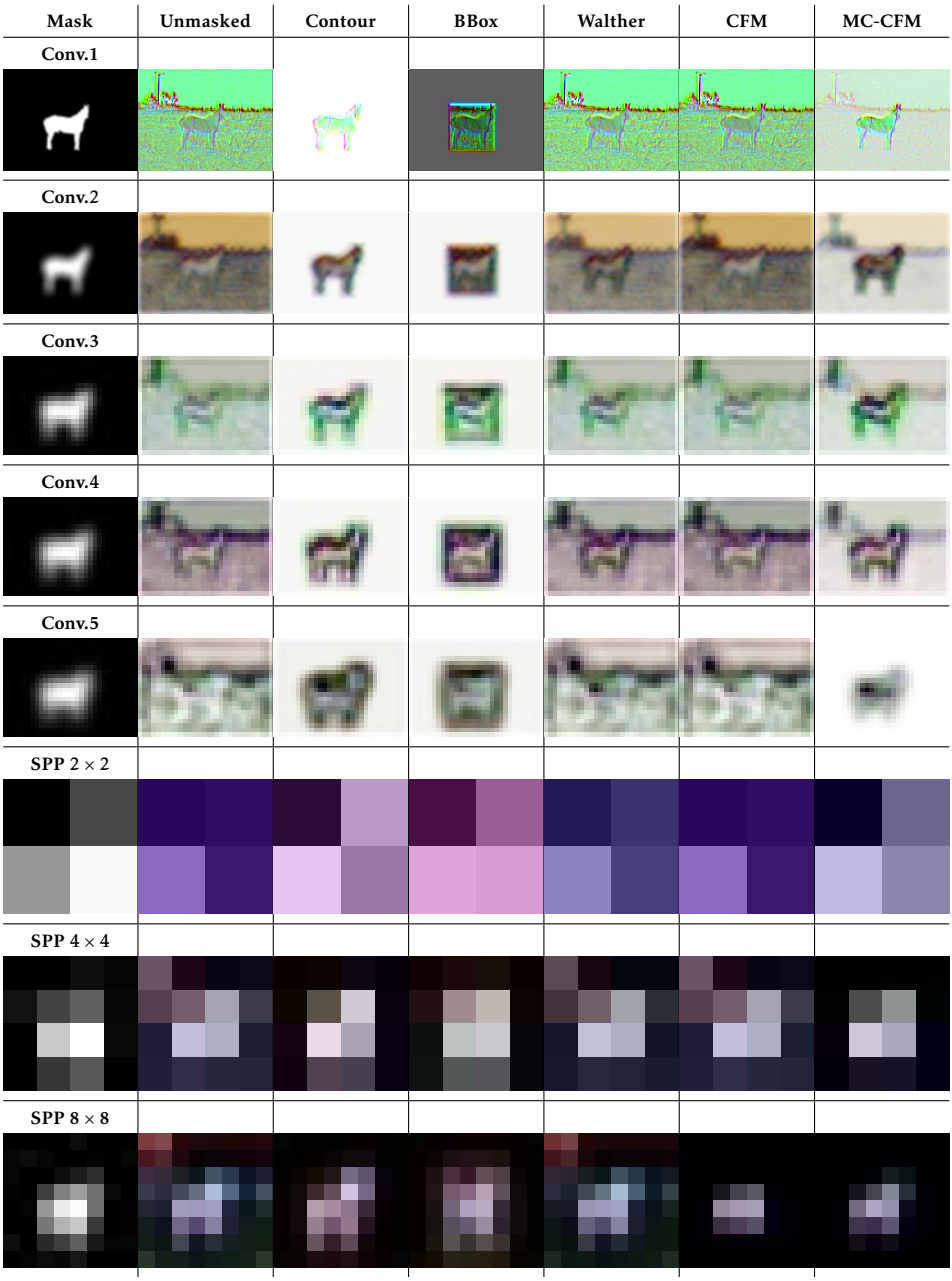

Figure 11.4: *Effect of masking on feature maps within a DNN. The input image is the one shown in figure 11.3. From left to right: The attention mask propagated through the network. The unmasked resulting activation pattern. Activation pattern when using a binary mask based on the object contour and bounding box, respectively. The masking method described in (Walther and Koch, 2006), convolutional feature masking (CFM) (Dai et al., 2015) and finally multi-layer continuous-valued convolutional feature masking (MC-CFM) (Wallenberg and Forssén, 2017). The false-colour images show a linear resampling to RGB over all feature map dimensions and is meant to illustrate spatial differences in activation distribution.*

12

Eddie: an EVOR platform

In order to study recognition in an embodied setting, an embodiment (in this case, a hardware and software platform) must be designed and implemented. The platform constructed in the EVOR project (2009-2013, known as Eddie the Embodied, see figure 12.1), is an example of such a system. This chapter will describe the structure and functionality of this system as it relates to experiments carried out during and after the project.

12.1 Hardware description

The principle behind the construction of the Eddie platform is simplicity of design, control and usage. Since the computer vision aspects rather than motor control were the focus of the project, the platform was designed such that the cameras were rigidly mounted onto a "head" with a fast pan-tilt unit to re-orient the entire assembly. The components, and their placement on the hardware rig can be seen in figure 12.2. The hardware consists of

- an aluminium head-and-neck construction with mount points for multiple cameras
- twin CCD cameras (Point Grey FL2G-13S2C-C) equipped with multiple sets of wide-angle optics and a stereo baseline of 120 mm
- a fast pan-tilt unit (Directed Perception PTU D-46-17.5) used to orient the head
- a structured light camera system for range estimation (used for calibration)
- a speaker system for providing audible user feedback through speech synthesis.

Figure 12.1: *Eddie the Embodied, a robotic platform designed to study learning, recognition and interaction in an embodied setting. Left: Eddie in 2009, right: Eddie in 2012 with re-positioned speakers and Kinect mounted.*

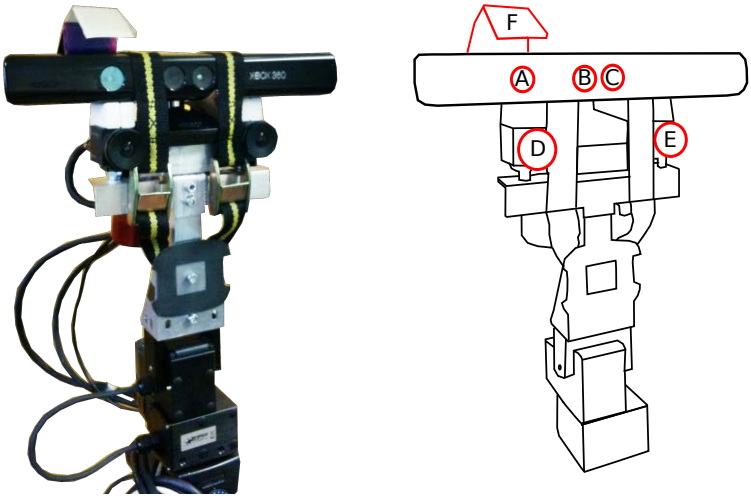

Figure 12.2: *System hardware.* **(A):** *structured light pattern emitter,* **(B):** *colour camera,* **(C):** *NIR camera,* **(D):** *right wide-angle camera,* **(E):** *left wide-angle camera,* **(F):** *structured light pattern diffusor. The speakers are not visible in these images.*

Figure 12.3: *Eddie's peripheral and foveal vision. Top row: left and right low-resolution peripheral views at* 320×240 *pixels. Middle row: target region with penguin in peripheral resolution (outer) and foveal resolution (inner). Bottom row: left and right high-resolution views at* 1280×960 *pixels from which foveal views are extracted.*

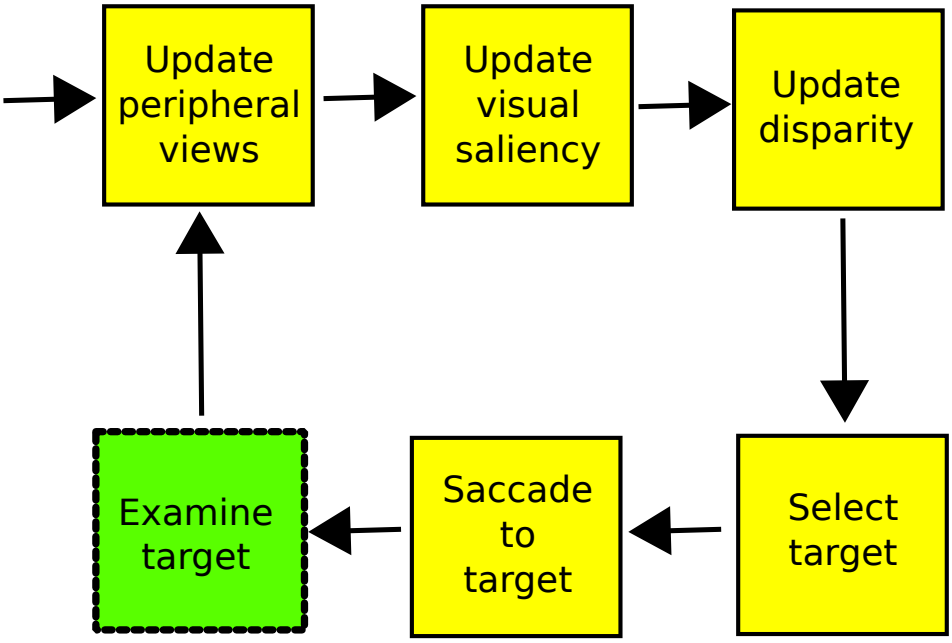

Figure 12.4: *Illustration of the attention-fixation loop. The system alternates between updating information about its surroundings, selecting salient targets using ICL and CtF-BFP and examining these, attempting to either recognise or learn them. A more detailed illustration of the examination procedure (dashed) can be found in figure 12.5.*

12.2 Software control structure during the EVOR project

The actions of the Eddie platform are centered around an *attention-fixation-recognition* loop. Using the wide-angle stereo cameras, the attention system searches for salient objects in view, and then attempts to recognise each object. The system also maintains a record of previously seen objects and fixation locations, so that the positions and types of objects can be verified after the initial identification. The structure of this attention loop is illustrated in figure 12.4.

12.2.1 Attention and visuomotor control

In unrectified wide-angle imagery, angular resolution decreases when moving away from the principal point. This means that objects in the periphery will be of both lower resolution and subject to significant shape distortions. The central region of the image, however, has maximum angular resolution, and little shape distortion. Therefore, in order to observe targets with high angular resolution

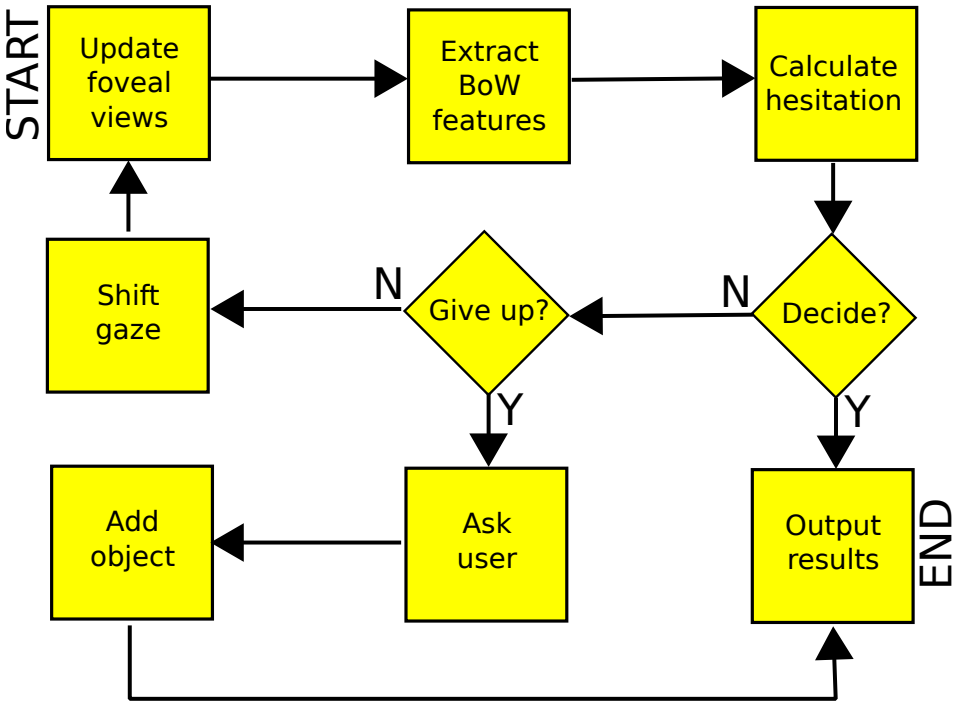

Figure 12.5: *Illustration of object recognition and learning. Upon fixating a target, the system captures a pair of high-resolution foveal views and computes BoW histograms of SIFT features. Classification is then attempted, and is considered a success or failure depending on the value of the confidence measure. If classification fails, a new pair of images from a slightly different viewpoint is added to the BoW and the confidence recalculated. The change in confidence over time is used to measure the hesitation of the system. While confidence is insufficient, more and more foveal views are added. This is repeated until either hesitation is sufficiently small (the object cannot be classified with confidence), or a set maximum number of frames have been captured. If this occurs, the user is prompted to identify the object, and the recorded features and location are added to the object memory.*

and low distortion, while retaining a wide field of view, the cameras need to be re-oriented to align with the target. In the current design, the cameras are fixed relative to the head, similar to a bird with frontally facing eyes. Thus, saccade motions are performed by moving the entire head. The system works with two different image resolutions, a low-resolution (320×240 pixels) *peripheral* view, used for attention and correspondence estimation, and a high-resolution *foveal* view of variable size at four times the angular resolution of the peripheral view.

The visual attention system consists of three parts, a change detection algorithm, a static saliency detector and an inhibition-of-return function. These are

all applied to the left peripheral view. The change detection algorithm is based on differences between static keyframes, and is used to detect the appearance or disappearance of objects in the visual field. The static saliency detector used is the *incremental coding length* (ICL) Hou and Zhang (2008) (see chapter 6). In addition to this, an inhibition map generated from previously attended locations in pan-tilt space is used to modulate the resulting saliency map. Maxima in the resulting target map are then used to select a fixation location, saccade motion and a *region of interest* (ROI) size. After the saccade motion, this region is extracted in high resolution from the left camera view. Vergence of the high-resolution regions is achieved by adjusting the ROI in the right image according to the correspondence map estimated by CtF-BFP (see section 5.2.3). Multiple high-resolution ROI views are then captured from both cameras, each with a small displacement around the fixation location (see section 12.2.2). This is done to reduce noise and promote identification of features that can be reliably extracted.

12.2.2 Learning and recognition

The Eddie platform uses a Bag-of-Words representation of objects with a pre-trained vocabulary (as described in chapter 9). Objects are stored in memory as BoW histograms, and each object is associated with a class name provided by the user. In the current implementation, SIFT features (Lowe, 2004) are extracted from the high-resolution foveal views, and then accumulated over all foveal views captured during the fixation. After each pair of foveal views, confidence and hesitation are evaluated according to (Wallenberg and Forssén, 2010a) (see part II, paper B), and the resulting confidence and hesitation values determine whether or not to make a decision about object identity. If no decision can be made due to low confidence or high hesitation, the user is asked to identify the object. All features extracted during the fixation are then added to the object memory associated with the user-specified class. The system currently uses a 7-NN classifier and a vocabulary of size 8000 prototype features. The object decision process is illustrated in figure 12.5.

12.3 Wide-angle stereo calibration and tuning

In order to make the Eddie rig into a useful wide-angle stereo system, it is necessary to determine what the appropriate methods and parameters for the intended application are. In this section, the calibration and tuning procedure used in (Wallenberg and Forssén, 2012) (see part II, paper D) is described, along with an expanded description of the point-point mappings and weighting scheme used.

12.3.1 Point-to-point mappings

Before automatic tuning of the wide-angle stereo system, the geometry of the pan-tilt camera setup must be calibrated. In order to do this, the mapping between the Kinect's measurements and points observed in the other cameras must be established.

Assuming that a 3D point $\mathbf{x}_k$ is visible in all cameras, the mapping from the projection $\mathbf{u}_{kAi}$ in the inverse depth image $d(\mathbf{u})_{Ai}$ generated by camera A at pan-tilt position i to another projection $\mathbf{u}_{kBj}$ in the image plane of another camera B at pan-tilt position j can be expressed as

$$\begin{aligned}\mathbf{u}_{kBj} &= \mathbf{K}_B \mathbf{f}_B \left[\mathbf{p}\left(\mathbf{R}_B^T\left(\mathbf{R}_j^T\left(\mathbf{x}_k - \mathbf{t}_0\right) + \mathbf{t}_0 - \mathbf{t}_B\right) + \mathbf{t}_B\right)\right], \quad \text{where} \\ \mathbf{x}_k &= \mathbf{R}_0 \mathbf{R}_i \left((\alpha d(\mathbf{u}_{kAi}) + \beta)^{-1} \mathbf{R}_A \left(\mathbf{f}_A^{-1}\left[\mathbf{K}_A^{-1}\mathbf{u}_{kAi}\right] - \mathbf{t}_A\right) + \mathbf{t}_A - \mathbf{t}_0\right) + \mathbf{t}_0. \end{aligned} \tag{12.1}$$

Here, $(\mathbf{R}_A, \mathbf{t}_A)$ and $(\mathbf{R}_B, \mathbf{t}_B)$ describe the positions and orientations of cameras A and B relative to some world coordinate system. $(\mathbf{K}_A, \mathbf{f}_A())$ and $(\mathbf{K}_A, \mathbf{f}_A())$ are the intrinsics and lens distortion parameters of the cameras and $\mathbf{R}_0$ and $\mathbf{t}_0$ describe the position and orientation of the pan-tilt axes. The rotation matrices $\mathbf{R}_i$ and $\mathbf{R}_j$ describe the two pan-tilt positions i and j. The α and β parameters define the mapping from inverse depth (output by the Kinect) to metric distance along the optical axis of camera A. The mapping $\mathbf{p}()$ denotes a projection normalisation operation such that the result is an actual point in the normalised image plane, rather than being only projectively equivalent to one.

If the origin and reference orientation are chosen such that the origin is at the optical center of the NIR camera, and such that the coordinate system is aligned to the normalised image plane, then $\mathbf{R}_0 = \mathbf{R}_A = \mathbf{I}$ and $\mathbf{t}_A = \mathbf{0}$. The point-to-point mapping can then be expressed as

$$\begin{aligned}\mathbf{u}_{kBj} &= \mathbf{K}_B \mathbf{f}_B \left[\mathbf{p}\left(\mathbf{R}_B^T\left(\mathbf{R}_j^T\left(\mathbf{x}_k - \mathbf{t}_0\right) + \mathbf{t}_0 - \mathbf{t}_B\right) + \mathbf{t}_B\right)\right], \quad \text{where} \\ \mathbf{x}_k &= \mathbf{R}_i \left((\alpha d(\mathbf{u}_{kAi}) + \beta)^{-1} \mathbf{f}_A^{-1}\left[\mathbf{K}_A^{-1}\mathbf{u}_{kAi}\right] - \mathbf{t}_0\right) + \mathbf{t}_0. \end{aligned} \tag{12.2}$$

Using this mapping, the transfer errors of known points can be calculated for all cameras and pan-tilt positions. This is what provides the cost function used in calibration. If, as stated in section 4.2.2, an image plane distortion model is used, the order of the mappings $\mathbf{K}_A$, $\mathbf{f}_A()$ and $\mathbf{K}_B$, $\mathbf{f}_B()$ are reversed.

12.3.2 Error variance propagation and weighting

Since the calibration procedure relies on detecting points in a calibration pattern in images from several *different* cameras, the effect of errors in these processes should be taken into account. The Kinect is used to reconstruct the 3D positions of calibration points, which are then mapped between cameras and pan-tilt positions, and these are then used to compute camera parameters and poses. The different properties of the cameras, such as different fields of view and resolutions, and the 3D position of each point affects its sensitivity to measurement errors, and thus its reliability. A useful weighting scheme should thus take this into account in the calibration procedure.

The purpose of the error variance propagation procedure is to determine the effects of measurement errors in position and inverse depth, in order to determine the variance of the resulting *transfer error* when mapping these measurements between cameras and pan-tilt positions. The estimated standard deviation

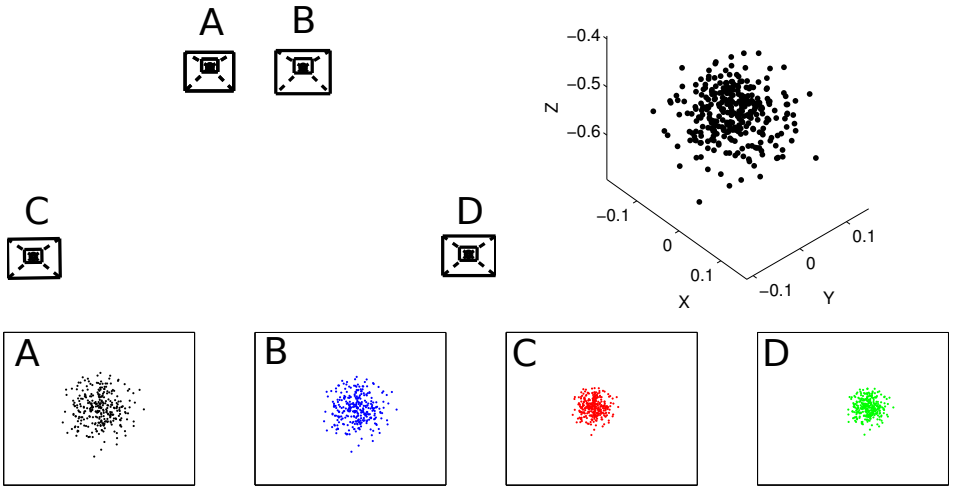

Figure 12.6: *Illustration of propagation of synthetic measurement errors in pixels ($\epsilon \sim N(0, 100)$) and inverse depth ($\epsilon \sim N(0, 50)$) from camera A (Kinect NIR camera) to 3D and other cameras B (Kinect colour camera), C (left wide-angle stereo camera) and D (right wide-angle stereo camera). Top left: estimated camera placements. Top right: 3D points calculated from position and inverse depth. Bottom row: image-plane projections of the resulting points.*

of this error is then used to normalise the variance of all transfer errors, thus achieving a weighting scheme where influence is inversely proportional to error variance. An illustration of the error variance propagation can be found in figure 12.6.

Since measurements of 3D points are obtained by backprojection from the inverse depth images, it is from these the error variances are propagated. In order to obtain an estimate that does not require a closed-form expression for the inverse leans distortion (which is not possible in some cases), the effects of lens distortion are not included in the variance propagation calculations. While significant lens distortion effects are present in the wide-angle imagery, they are of the *barrel* type (see figure 4.3). This means that image-plane deviations are "squashed" in the image, and that disregarding the distortion will result in an overestimation of the sensitivity to perturbations. This can therefore be considered a conservative measure of reliability, since the actual errors will be smaller than those predicted by the model.

Under the appropriate assumptions (see (Wallenberg and Forssén, 2012) for details, part II, paper D), the variance of the sum of transfer error and localisation error along each coordinate axis can be expressed as the sum of their individual variances. The inverse standard deviation of this error sum is then used to weight

the resulting residual in the desired way.

12.3.3 Calibration procedure

The actual calibration procedure is carried out in a predefined sequence in order to properly obtain starting values of parameters in a robust way. First, intrinsics and and distortion parameters are estimated using publicly available implementations of the methods described in (Zhang, 2000) and (Heikkilä and Silven, 1997). Once this has been done, inverse depth conversion parameters for the Kinect are estimated. With these parameters in place, reconstruction of 3D points from the Kinect's inverse depth image is possible. Reconstructed points on the calibration pattern are then used to find the relative poses of the two Kinect cameras and the left and right wide-angle cameras. The entire assembly is then rotated around the pan and tilt axes, and the intersection of these axes is estimated. Finally, the initial estimates of all parameters are refined using the images captured at all pan-tilt positions.

12.4 Later experiments

After the EVOR project, the Eddie platform was also used in various experiments popular science demonstrations. Change detection using *Gaussian mixture models* (GMM:s) was added for detection of new targets. An improved target segmentation based on (Mishra and Aloimonos, 2009) was added for experiments on attention masking in deep networks. The classifier used on the platform for these experiments was a pre-trained deep neural network with frozen convolutional layers from the `VGG-F` network (Chatfield et al., 2014), with an added *spatial pyramid pooling* (SPP) layer (He et al., 2014) and fully-connected layers trained on the PASCAL-Context dataset (Mottaghi et al., 2014). The details of these experiments can be found in (Wallenberg and Forssén, 2017).

13

Concluding Remarks

Studying system applications like embodied recognition requires insight into many research areas within computer vision, and also requires methods from these to be integrated into practically usable systems. Perhaps more so than in other areas, solutions change rapidly and the interdependence of multiple functionalities complicates analysis, comparison and development. As techniques need to be relatively mature before inclusion into a complex system, there is often a considerable gap between the latest methods and those used in embodied systems. However, the practical considerations within this area mean that it is, in the author's view, well-suited suited to bridging the gap between frontier research and mainstream applications. In order to be successful, an embodied recognition system needs to be:

- *Interactive* in the sense that it can react to real-world occurrences fast enough to enable meaningful interactions with its surroundings and users.
- *Scalable* and *adaptive* in that the mechanisms for recognition must be able to maintain specificity and robustness with increasing numbers of object categories and also able to incorporate new observations into its object models. This is similar to Piaget's concept of adaptation through *assimilation* and *accommodation* (for a description of these concepts, see for instance (Solso et al., 2008)).
- *Parallel and holistic* in that the recognition problem should be addressed as more than a simple classification or matching problem, incorporating dynamic aspects of embodiment such as view selection and deferred decisions.

Introducing these considerations already in the design phase and incorporating methods for adaptive target observation, segmentation from multiple cues, automatic stereo tuning, feature selection and attentional masking into modern machine learning frameworks has the potential to improve future embodied recognition systems. While much of this remains to be realised, the insights gained in this thesis will no doubt be of use in further investigations into these issues.

Bibliography

Radhakrishna Achanta and Sabine Süsstrunk. Saliency detection using maximum symmetric surround. In *Proceedings of The International Conference on Image Processing*, 2010.

Radhakrishna Achanta, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC Superpixels, EPFL Technical Report 149300. Technical report, EPFL, 2010.

Alexandre Alahi, Raphaël Ortiz, and Pierre Vandergheynst. FREAK: Fast Retina Keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, 2006.

Serge Beucher and Christian Lantuéjoul. Use of watersheds in contour detection. In *International workshop on image processing, real-time edge and motion detection*, 1979.

M. Blatt, S. Wiseman, and E. Domany. Superparametric clustering of data. *Physical Review Letters*, 1996.

Magnus Borga. Canonical correlation: a tutorial. Technical report, Linköping University, 2001. `http://www.imt.liu.se/people/magnus/cca/tutorial/tutorial.pdf`.

A. Borji, D. N. Sihite, and L. Itti. What/where to look next? modeling top-down visual attention in complex interactive environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A - Systems and Humans*, 2014.

Leo Breiman. Random forests. 2001.

J. Campbell. *Film and Cinema Spectatorship: Melodrama and Mimesis*. Wiley, 2005.

Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional networks. In *Proceedings of the British Machine Vision Conference*, 2014.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. *Computing Research Repository*, 2016. URL http://arxiv.org/abs/1602.07576.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3), 1995.

Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 2012.

Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Céderic Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning for Computer Vision*, 2004.

Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *Computing Research Repository*, 2013. URL http://arxiv.org/abs/1310.1531.

Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22(1), 1973.

M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting, with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119–130, 1988.

James J. Gibson. *The theory of affordances*. Wiley, 1977.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016. URL http://www.deeplearningbook.org.

Gösta H. Granlund. An associative perception-action structure using a localized space variant information representation. In *AFPAC*, 2000.

D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1989.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

Trevor J. Hastie, Robert John Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction, 2nd Ed.* Springer series in statistics. Springer, New York, 2013. ISBN 978-0-387-84857-0.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computing Research Repository*, 2015. URL `http://arxiv.org/abs/1512.03385`.

Donald Hebb. *The Organization of Behavior*. Wiley, 1949.

Janne Heikkilä and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1997.

Tin Kam Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*. IEEE Computer Society, 1995.

Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

Xiaodi Hou and Liquing Zhang. Dynamic visual attention: Searching for coding length increments. In *NIPS*, 2008.

D.H. Hubel and T.N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *J Physiol.*, 3:574–591, 1959.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Computing Research Repository*, 2015. URL `http://arxiv.org/abs/1502.03167`.

L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov 1998.

P. Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:241–272, 1901.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc., 2015.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 1972.

Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7), 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

M.F. Land and D.E. Nilsson. *Animal eyes*. Oxford animal biology series. Oxford University Press, Incorporated, 2002.

Michael F. Land. Eye movements and the control of actions in everyday life. *Progress In Retinal And Eye Research*, 25(3), May 2006.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, number 11, pages 2278–2324, 1998.

S Leutenegger, M Chli, and R Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

Maxime Lhuillier and Long Quan. Match propagation for image-based modelling and rendering. 2002.

Maxime Lhuillier and Long Quan. Image-based rendering by joint view triangulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

Charles Loop and Zengyou Zhang. Computing rectifying homographies for stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, 1999.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943.

Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

Ajay K. Mishra and Yiannis Aloimonos. Active segmentation with fixation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

Dan-Eric Nilsson. The evolution of eyes and visually guided behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2009.

Mustafa Özuysal, Pascal Fua, and Vincent Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 1901.

David I. Perrett and Mike W. Oram. Neurophysiology of shape processing. *Image and Vision Computing*, 1993.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *Computing Research Repository*, 2014. URL `http://arxiv.org/abs/1403.6382`.

John H. Reynolds and David J. Heeger. The normalization model of attention. *Neuron*, 2009.

Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 1999.

Sanna Ringqvist, Pelle Carlbom, and Marcus Wallenberg. Classification of Terrain using Superpixel Segmentation and Supervised Learning. In *Proceedings of SSBA 2015 Symposium on Image Analysis*, 2015.

Juan J. Rodriguez, Ludmila I. Kuncheva, and Carlos J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

F Rosenblatt. The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. *Psychological Review*, 1958.

Edward Rosten, Reid Porter, and Tom Drummond. FASTER and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 1988.

Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 2002.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *Computing Research Repository*, abs/1503.03832, 2015. URL `http://arxiv.org/abs/1503.03832`.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Computing Research Repository*, 2016. URL `http://arxiv.org/abs/1605.06211`.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

R.L. Solso, O.H. MacLin, and M.K. MacLin. *Cognitive psychology, 8th Ed.* Allyn and Bacon, 2008.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

J.V. Stone. *Vision and Brain: How We Perceive the World*. MIT Press, 2012.

Taffee T. Tanimoto. IBM internal report. Technical report, 1957.

A M Treisman and G Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 1980.

Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*. Springer Verlag, 2000.

Marcus Wallenberg. A Simple Single-Camera Gaze Tracker using Infrared Illumination. In *Proceedings of SSBA 2009 Symposium on Image Analysis*, pages 53–56, 2009.

Marcus Wallenberg. *Components of Embodied Visual Object Recognition: Object Perception and Learning on a Robotic Platform*. Linköping University Electronic Press, 2013. ISBN 978-91-7519-564-3. Licentiate Thesis no. 1607.

Marcus Wallenberg and Per-Erik Forssén. Embodied Object Recognition using Adaptive Target Observations. *Cognitive Computation*, 2(4):316–325, 2010a.

Marcus Wallenberg and Per-Erik Forssén. A Research Platform for Embodied Visual Object Recognition. In *Proceedings of SSBA 2010 Symposium on Image Analysis*, pages 137–140, 2010b.

Marcus Wallenberg and Per-Erik Forssén. Teaching Stereo Perception to YOUR Robot. In *Proceedings of the British Machine Vision Conference*, 2012.

Marcus Wallenberg and Per-Erik Forssén. Automatic Stereo Tuning for YOUR Robot. In *Proceedings of SSBA 2013 Symposium on Image Analysis*, 2013.

Marcus Wallenberg and Per-Erik Forssén. Improving Random Forests by correlation-enhancing projections and sample-based sparse discriminant selection. In *Computer and Robot Vision*, 2016.

Marcus Wallenberg and Per-Erik Forssén. Attention Masking for Pre-trained Deep Networks. (Submitted), 2017.

Marcus Wallenberg, Michael Felsberg, Per-Erik Forssén, and Babette Dellen. Channel Coding for Joint Colour and Depth Segmentation. In *Proceedings of the DAGM Symposium on Pattern Recognition*, 2011a.

Marcus Wallenberg, Michael Felsberg, Per-Erik Forssén, and Babette Dellen. Leaf Segmentation using the Kinect. In *Proceedings of SSBA 2011 Symposium on Image Analysis*, 2011b.

Dirk Walther and Christof Koch. Modeling attention to salient proto-objects. *Neural Networks*, 2006(19):1395–1407, 2006.

J. Wang, A. Borji, C.-C. J. Kuo, and L. Itti. Learning a combined model of visual saliency for fixation prediction. *IEEE Transactions on Image Processing*, 2016.

P.J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University, 1974.

Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

Part II

Publications

Papers

The articles associated with this thesis have been removed for copyright reasons. For more details about these see:

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-132762