

Outlier Removal in Real-time Object Recognition and Pose Estimation

MANG SHAO

Electrical and Electronic Engineering

Imperial College
London

SUPERVISED BY: DR. TAE-KYUN KIM

This dissertation is submitted for the degree of

Doctor of Philosophy

February 2017

Abstract

Outlier removal algorithms aim to detect and remove abnormal or negative data which sufficiently differ from training samples. Since most object recognition or pose estimation methods involve a hypothesise-and-test scheme, especially for large-scale or real-time problem, outlier removal algorithms can be essential for desirable performance. Unlike domain adaptation or transfer learning, outlier removal algorithms usually do not have prior knowledge of negative samples during training. Rather than having a universal solution, performing outlier removal algorithm usually depends on the task and the applied machine learning technique.

In this thesis, we investigate the application of outlier removal algorithm in object recognition and pose estimation problems. Specifically, we classify them into three types and investigate one application from each: a comparative study for object recognition in video as the distance-based approach; a new grouped outlier removal method for robust ellipse fitting as the registration-based approach; and a novel real-time background-aware 3D texture-less pose estimation method as the learning-based approach.

The comparative study is centred around using a wide choice of spatial and temporal consistencies to remove outlier feature points. State-of-the-art techniques are classified, implemented under a unified framework, and empirically evaluated with a newly collected museum dataset. For geometric cues, we find that 3D object structure learnt from a training video dataset improves the average video classification performance dramatically. By contrast, for temporal cues, tracking visual fixation among video sequences has little impact on the accuracy, but significantly removes background fea-

ture points and reduces memory consumption. Furthermore, we propose a method that integrates these two cues to exploit the advantages of both.

Then, we presents a registration-based outlier removal method which is capable of fitting ellipse in real-time under high outlier rate, based on the phenomenon that outliers generated by ellipse edge point detector are likely to appear as groups due to real-world nuisances, such as under partial occlusion or illumination change. To confront the grouped outliers while maintaining the fitting efficiency, we introduce a proximity-based ‘split and merge’ approach to cluster the edge points, followed by a breadth-first outlier removal process. The experiment shows that our algorithm achieves high performance under a wide range of outlier ratio and noise level with various types of realistic nuisances.

An outlier-aware extension of randomised decision forest is proposed and applied to real-time 3D object pose estimation problem based on typical template matching methods. A set of templates uniformly covering the pose space is generated during training and the nearest neighbour to query point is found during testing. Since the amount of data raised from the background is in the orders of magnitude more than foreground during testing, it is desirable to reject the background early to save computational power as much as possible. Hence the conventional randomised decision tree is modified to a ternary tree, where each node, apart from the original children, contains an additional ‘background rejection’ node. During testing, the query data far from training samples will be detected and rejected along the propagation down the trees. Furthermore, we propose the application of ‘fuzzy decision’ instead of binary when training the decision forest to raise the tolerant to ambiguous data samples so that the sample near the decision boundary will be assigned to both left and right child nodes. Our approach is also scalable to large datasets, since the tree structure naturally provides a logarithm time complexity to the number of objects. Finally, we further reduce the validation stage with a fast breadth-first scheme. The results show

that our approach outperforms state-of-the-arts on efficiency while maintaining comparable accuracy.

Keywords

outlier removal, object recognition, 3D object pose estimation, ellipse fitting, randomised decision tree, template matching.

Declaration of Originality

I, Mang Shao, declare that this thesis is my own work and information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.

Main contents have been published in the following listed papers. For these works, I made the major contribution in design, implementation and experiments.

- **Mang Shao, Danhang Tang, Yang Liu and Tae-Kyun Kim**, "A comparative study of video-based object recognition from an egocentric viewpoint", *Neurocomputing*, 2015
- **Mang Shao, Yoshihisa Ijiri and Kosuke Hattori**, "Grouped outlier removal for robust ellipse fitting", in *Proc. of Machine Vision Applications (MVA)*, Tokyo, Japan, 2015
- **Mang Shao, Danhang Tang, Tae-Kyun Kim**, "Real-time Background-aware 3D Textureless Object Pose Estimation ", arXiv preprint arXiv:1907.09128.

Apart from the above, I co-authored the following publications that, although out of the main scope, are briefly mentioned in the literature review and future directions.

- **Yang Liu, Minh Hoai, Mang Shao and Tae-Kyun Kim**, "Latent Bi-constraint SVM for Video-based Object Recognition", arXiv preprint arXiv:1605.09452.

Mang Shao

28/02/2017

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Mang Shao
28/02/2017

Acknowledgements

First and foremost, I wish to thank my advisor, Dr. Tae-Kyun Kim, director of Computer Vision and Learning Lab at the Imperial College London (ICVL), for his continuous support in my Ph.D study and related research.

Dozens of people have helped and taught me immensely at the ICVL. Youngkyoon Jang and Yang Liu were of tremendous help when I started my Ph.D study. I would also like to express my gratitude to Danhang Tang for his inestimable advice and support. I would also like to thank Wenhong Luo, Chao Xiong, Zhiyuan Shi, and everyone else at the ICVL who influenced my work.

Last but not the least, I would like to thank my parents for supporting me throughout writing this thesis and my life in general.

Mang Shao
28/02/2017

CONTENTS

LIST OF FIGURES	xii
LIST OF ALGORITHMS	xix
LIST OF TABLES	xx
CHAPTER 1	
INTRODUCTION	1
1.1 Problem Definition	3
1.2 Thesis Structure and Contributions	7
CHAPTER 2	
A REVIEW OF OUTLIER REMOVAL METHODS	11
2.1 Motivation	11
2.2 What is outlier	13
2.3 Challenges	14
2.4 Related works	16
2.5 Aspects of Outlier Removal Method	18
CHAPTER 3	
OUTLIER REMOVAL IN VIDEO-BASED OBJECT RECOGNITION: A COMPARATIVE STUDY	35
3.1 Overview	36

CONTENTS

3.2 Approaches	38
3.3 Implementation	44
3.4 Dataset	48
3.5 Evaluation	52
3.6 Summary	60
CHAPTER 4	
REAL-TIME BACKGROUND-AWARE 3D TEXTURELESS OBJECT POSE ESTIMATION	63
4.1 Overview	63
4.2 Related work on 6-DoF pose estimation	65
4.3 Method	69
CHAPTER 5	
A GROUPED-OUTLIER-AWARE REGISTRATION-BASED METHOD FOR ROBUST ELLIPSE FITTING	87
5.1 Overview	88
5.2 Problem Setting and Preliminaries	90
5.3 Proposed Approach	93
5.4 Evaluation	97
5.5 Summary	100
CHAPTER 6	
CONCLUSION AND FUTURE WORKS	103
6.1 Conclusions	103
6.2 Future Works	105
CHAPTER 7	
APPENDIX	109
7.1 Method implementations	109

CONTENTS

BIBLIOGRAPHY

131

CONTENTS

LIST OF FIGURES

1.1	Applications of object recognition and pose estimation.	3
1.2	Examples of visual variations from a single object instance of our proposed V&A museum video-based object recognition dataset.	5
1.3	A class-generic object detector [Alexe et al., 2010] that discards non-object sliding windows in the image.	7
2.1	Toy examples of different outlier types.	13
2.2	Different pixel-region-of-interest in a positive dataset for object recognition, detection, semantic segmentation task.	15
2.3	Applications that utilise outlier removal methods and the research areas behind the techniques.	16
2.4	Computer vision applications that heavily utilise outlier removal.	18
2.5	Examples of data type versus relation type.	20
2.6	A one-class classifier applied to a toy example. The solid line represents the conventional classifier that distinguishes between ‘apple’ and ‘orange’, while dash line is the one-class classifier that detects the outlier that does not belong to any of classes.	22

2.7	Figure 2 in the paper of [Tax and Duin, 2004]. It shows a tightly trained data description on a banana shaped data set, a good description should cover all target data but includes no superfluous space. Outlier removal problem can make good of this boundary. They also applied a polynomial kernel with varying degrees. Support vectors are indicated by the solid circles, the dashed line is the description boundary.	23
2.8	A k-nearest neighbour algorithm applied to a toy example, where $k = 3$. An outlier is detected when the sum of distance of its nearest neighbours is significantly larger than other data points.	25
2.9	Early rejection of background queries in Chapter 4.	26
2.10	A simple example of registration-based outlier removal method (RANSAC). Assuming the point set contains inlier that can be fit to a line, this approach iteratively finds the best line that fits the most data points. All the data points that do not fit the given model are considered outliers.	29
2.11	Experiment list for methods with/without outlier removal approaches in Chapter 2.	32
3.1	Method categorisation and experimental setup.	38
3.2	Image-based methods selected from three state-of-the-art object recognition frameworks.	39
3.3	Toy example of set-based methods.	42
3.4	A toy example of trajectory matching methods based on feature tracking.	43
3.5	Toy example of 3D-based methods where each video is treated as an un-ordered image set.	44
3.6	Illustration of the collected dataset.	48

- 3.7 Full evaluation of the video classification results based on the precision-recall curves (all figures are best viewed in colour), including image-based methods (with voting), set-based methods, video-based methods and 3D-model based methods. 49
- 3.8 We divided our *V&A* dataset into 10 subsets by different types of nuisance. Each column represents one subset which contains 33 videos (one per class), whilst each row gives the results of a method. The numbers indicate the amount of successfully classified videos. 53
- 3.9 This table provides a coarse time consumption of each workflow of methods. The green colour indicates real-time, i.e. beyond 10 frames per second (FPS), or fast technique; The yellow colour indicates the method runs between 1 FPS to 10 FPS, or medium speed technique; The red colour indicates the method runs below 1 FPS, or slow technique. 54
- 3.10 3D object point clouds are reconstructed via structure-from-motion algorithms for 2D-to-3D geometric validation. 56
- 3.11 Histogram showing (a) the number of frames crossed per trajectory and (b) the number of SIFT features allocated per 3D point. (c) The remaining percentage of features after outlier removal via different methods. 58
- 3.12 Precision-recall curves with the hybrid method compared with the best method from each method category. 59
- 4.1 We generate synthetic dataset from object models that provided by LineMOD dataset. The left figure shows a sample procedure of rendering templates on a hemisphere of several radii. Here we use four modalities in this work, from left-top: colour gradient, surface normal, hue colour and depth. 71

- 4.2 Visualisation of our proposed pipeline. At each candidate sliding window, we extract LineMOD feature descriptor and pass to the decision tree. At each node, the extracted features are examined by a preemptive background rejector, the candidate enters full validation stage only if it passes all the rejectors. This process significantly save the time cost on background noises. Finally, we further accelerate the validation stage with a fast breadth-first scheme, inspired by [Nistér, 2005a]. 72
- 4.3 Breath-first preemptive scheme for leaf validation speed up. The templates are equally split into small chunks to alleviate the time cost from validating bad candidates. 77
- 4.4 In this sample frame, with our proposed preemptive background rejector up to 97% negative bounding boxes are filtered before reaching the forest leaf nodes. The green region in left image indicates the locations that enter the validation stage; the right image shows the tree depth when background locations are rejected, dark blue indicates the region is either rejected due to out-of-depth-range or pass the validation stage, light blue to yellow indicates whether the locations are rejected early or late in the forest. As nearer to the ground truth or ambiguous objects the location is more likely to not be rejected earlier. 81
- 4.5 Top: max decision tree depth versus runtime and accuracy. Bottom: fuzzy factor ζ versus runtime and accuracy. 82
- 5.1 A challenging case for proximity-based outlier removal method. Type (a) outliers cannot be filtered by simple proximity (e.g. k-Nearest Neighbour) check. Type (b) is even more difficult since they are connected with the inlier contour. 90

5.2	This figure shows an example that the value of the smallest generalised eigenvalue λ decreasing drastically after excluding the outlier subsets. The point set contains two ‘outlier’ subsets, (a), (b) and (c) show three cases during the iteration, while (a) does not exclude any outlier subset, (b) excludes one of them and (c) excludes all.	95
5.3	Evaluation result of 3 methods on synthetic and realistic datasets (best view in color).	98
7.1	Figure 1 in the paper of [Lowe, 2004a]. Figure (a) shows how multi-scaled DoGs are calculated; figure (b) shows the approach for finding local maxima/minima of DoG.	110
7.2	Example of SIFT descriptor on a sample image of our proposed dataset.	113
7.3	A brief workflow of Bag-of-Words approach.	114
7.4	A toy example from Figure 1 of the work [Lazebnik et al., 2006]. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, the image is subdivided at three different levels of resolution. At each level of resolution, a histogram is formed and weighted. The final representation takes the concatenation of all histograms.	115
7.5	An experiment conducted in the work [Muja and Lowe, 2014]. Search efficiency for data of varying dimensionality, with data set of size 100K. (a) uses data with no correlations (random vectors), (b) uses real-world image descriptors.	116
7.6	An example of kd-tree from work [Muja and Lowe, 2014]. The data is split until each leaf node has one data point.	117
7.7	A camera calibration board.	119
7.8	From [Lepetit et al., 2009]. Examples of reprojection of the models on real images.	121

LIST OF FIGURES

- 7.9 An example of local points tracking. The square boxes are the local points in current frame, the dots are the local points in previous frames. 125
- 7.10 Reconstructed 3D point cloud from our proposed dataset. 127

LIST OF ALGORITHMS

1	Proximity-based Point Clustering	94
2	Preemptive searching	96

LIST OF TABLES

- 4.1 Accuracy and average time per frame for the whole pipeline with 1, 5, 20 and 50 trees. Our approach is several times faster than the state-of-the-art approaches with comparable accuracy. T_Tree, T_Valid and T_Total are the time cost per frame from decision tree, leaf validation and overall respectively. 78
- 4.2 Accuracy and average time per frame for multiple object (5 trees) and its comparison with state-of-the-art approaches. 79
- 4.3 Accuracy and average time per frame for self-comparison, evaluated on 13 objects. Each listed improvements lead to significant increases in performance. Baseline: kd-tree with Randomised decision forest (RF) by [Muja and Lowe, 2014]; improv. 1: Fuzzy split function (FZ); improv. 2: randomised ternary tree with extra rejector nodes (RN); improv 3.: Breath-first search in leaf validation (LV). 79

CHAPTER 1

INTRODUCTION

CONTENTS

1.1 Problem Definition	3
1.2 Thesis Structure and Contributions	7

With the recent breakthroughs in augmented reality (AR) technology, living in a world that is a blend of the virtual and real seems a tangible future. Super-imposing virtual objects, such as projecting navigation details onto windscreen becomes a common practice in many AR applications, where object recognition and pose estimation algorithms play an fundamental role.

Microsoft Hololens™ as shown in Figure Fig. 1.3 (a), also known under development as Project Baraboo, is a pair of mixed reality head-mounted display smartglasses released in 2016. Through the glasses, the user can place, scale or model a variety of 3D objects around them, which provides great convenience to designers, gamers, businessmen and more. One of the most important technologies behind it is its ability to

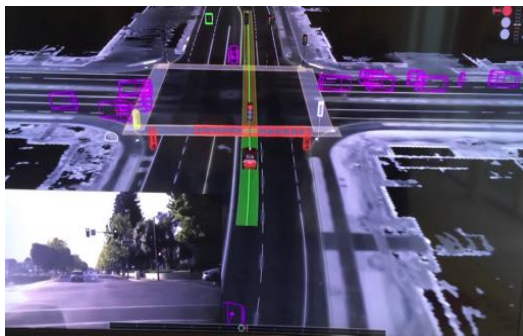
capture and recognise the scene and 3D objects around the user. Figure Fig. 1.3 (b), 'Search by image', allows user to search for related images by uploading an image. This is achieved by analysing and comparing the submitted image with other images in the databases. The common algorithms include many techniques involved in object recognition tasks. In Figure Fig. 1.3 (c), Google's self-driving car is a promising fully self-driving technology that will handle all the driving without taking the wheel; it will have a big impact on improving road safety and mobility. While navigating, the sensors in the vehicle accurately detect pedestrians, cyclists, vehicles and more from a large distance on the fly. In the industry, monotonous tasks are labor-intensive. Figure Fig. 1.3 (d) shows a robot arm picking the workpieces in an automated factory. With a good recognition accuracy, a great amount of human efforts and resources will be saved. Despite the huge potential benefit, the development of most of the above mentioned applications are still ongoing and have not been reliable enough to reach commercially acceptance. This motivates us to further improve the performance of object recognition and pose estimation algorithms.



(a) Microsoft's HoloLens system



(b) Reverse image search



(c) Google self-driving car



(d) Bin-Picking Robot System

Figure 1.1: Applications of object recognition and pose estimation.

1.1 Problem Definition

1.1.1 Object recognition and Pose estimation

Humans can interpret a multitude of objects without almost any efforts, despite the possibility of image of the objects being varied in innumerable possibilities of identity-preserving image transformations, including changing position, size, angle of view, context and even shape deformation to some extent. Although the recognition mechanism in our brain remains poorly understood, scientists are doing their best efforts to mimic the human visual system in modern object recognition algorithms.

3D object recognition and pose estimation remains a difficult problem in the sense of pose accuracy and scalability despite many algorithms having been proposed in

the past few decades. Since there are 6 degrees of freedom (3 in translation and 3 in rotation), it has also been dubbed 6DoF pose estimation. Even if we quantise the pose space, an object still needs to render hundred or thousands of possible poses in order to have a satisfying accuracy. This naturally costs much more computational power compared to the object recognition problem, which explains how state-of-the-art 2D image recognition algorithms can scale to millions of object classes, whilst 6DoF pose estimation algorithms usually tackle only a few or tens of them.

To retrieve object identity and poses, the first challenge is to accommodate the extraordinary amount of visual nuisances that exhibit real-world scenarios. Even moderate scale, pose or light changes in an object could lead to significant changes of the pixel array, as illustrated in Figure Fig. 1.2. To compensate for the performance loss from complex nuisances, a naive solution is to collect a more complete description of objects during training. However, this is usually impractical, since manually collecting and labelling object data is labour-intensive, time consuming and error-prone.

In addition to object identity, pose estimation tasks usually require tens of thousands of training data for each object to uniformly cover the pose space. This grows exponentially with the pose degrees of freedom. For instance, to estimate the 3D pose of a rigid object, 6 degrees of freedom are needed. This is even more labour-intensive or simply infeasible for manual annotation.

Apart from labelling, since the search space grows exponentially with degrees of freedom, it is much more difficult to precisely locate a pose parameter. Also, in many algorithms, especially non-parametric ones, more training data usually leads to larger memory consumption, as well as slower runtime. This hinders applications that require real-time performance.

An economic and effective way of collecting training data for 3D objects is to render a large quantity of synthetic training images. Annotations can be obtained without ef-



Figure 1.2: Examples of visual variations from a single object instance of our proposed V&A museum video-based object recognition dataset.

fort with this method. However, the lack of background or ‘negative’ training samples usually leads to a high false positive rate during testing. Hence, one needs to be able to accurately and efficiently tell normal and negative data apart, whilst keeping in mind the discrepancy between synthetic positive training data and real positive data. On the other hand, the object of interest in many realistic scenarios can be very small such that almost all image regions are background clutters. Rapidly locating the region of interest with high recall rate before an expensive classification algorithm will vastly reduce

computation complexity. These highly motivate the improvement of outlier removal algorithms for overall performance.

1.1.2 Outlier Removal in Computer Vision tasks

Outlier removal is an indispensable component in many domains. In computer vision tasks, often assuming the available training data are ‘positive’, during testing, positive matching and negative (or outlier) detection are simultaneously conducted with machine learning techniques to achieve better recognition performance. The concept is closely related to ‘one-class learning’, but differs from Domain Adaptation or Transfer Learning, where partial novel data are available during training. In this thesis, we focus on using outlier removal techniques in various computer vision tasks to verify the positive data and/or remove the outlier. Also we study how these techniques can be significant in general machine learning systems.

Outlier removal methods have been extensively explored in many domains, such as medical diagnosis [Hauskrecht et al., 2013], spam detection [Idris et al., 2015], sensor network [Branch et al., 2013], astronomy catalogues [Dutta et al., 2007], data mining and so on. In the field of computer vision, they been applied to detect abnormal behaviours in video surveillance [Bouwman and Zahzah, 2014], live structure and motion estimation for 3D reconstruction [Szeliski, 2010], building large face detection dataset [Ng and Winkler, 2014b] and many more in object recognition, detection and pose estimation pipelines [Hao et al., 2013a, Beis and Lowe, 1997, Hinterstoisser et al., 2012b, Gordon and Lowe, 2006a]. A more detailed introduction will be presented in the next chapter.



Figure 1.3: A class-generic object detector [Alexe et al., 2010] that discards non-object sliding windows in the image.

1.2 Thesis Structure and Contributions

This thesis investigates how to adopt outlier removal in object recognition, ellipse fitting and pose estimation problem. We start by a general review of state-of-the-art outlier removal methods in Chapter 2. Chapter 3 covers a comparative study of how spatial and temporal information can be used for detecting outliers under a modern object recognition framework. Chapter 5 introduces a registration-based outlier removal method for ellipse fitting problem. A distance-based method is proposed in Chapter 4. Finally, Chapter 6 gives a conclusion and a plan for the future work.

Highlights of the main chapters are listed as below:

Chapter 2. A Review of Outlier Removal Methods

This chapter gives a general and timely review of state-of-the-art outlier removal methods. Judging through the machine learning techniques, these methods are categorised by their applications. This categorisation also lays out a structure for the following chapters.

Chapter 3. Outlier removal in Video-based Object Recognition: A Comparative Study

This chapter conducts a comparative study of how outliers removal methods are

applied under different object recognition methods for video-based rigid object instance recognition (VbOR). First, the diverse state-of-the-art VbOR techniques are categorised, extended and evaluated empirically using a newly collected video dataset which consists of complex sculptures in clutter scenes. During the experiments, we investigate how to utilise the geometric and temporal cues provided by egocentric video sequences to detect the outlier, and hence, improve the performance of object recognition. Based on the experimental results, we analysed the pros and cons of these methods and reached the following conclusions: for geometric cues, the 3D object structure learnt from a training video dataset improves the average video classification performance dramatically. By contrast, for temporal cues, tracking visual fixation among video sequences has little impact on the accuracy, but significantly accelerates the matching process by removing outlier feature points detected in the query frame. Furthermore, we proposed a method that integrated these two important cues to exploit the advantages of both.

The contributions for this chapter are summarised as below:

- A dataset named *Sculptures in Victoria and Albert (V&A) Museum dataset* is collected from an egocentric viewpoint, then processed for evaluating different types of video-based object recognition methods.
- Diverse state-of-the-art object recognition frameworks and their video-based extensions are surveyed and evaluated.
- A hybrid solution from object recognition frameworks is further proposed to combine the advantages of both temporal and spatial cues.

Chapter 4. Real-time Background-Aware 3D Textureless Object Pose Estimation

An outlier removal method is proposed and applied to a 3D object pose estimation problem. A typical solution for 3D pose estimation is template matching. A set of templates uniformly covering the pose space is generated during training, and the nearest

neighbour to query template is found during testing. Using random forest to speed up this searching process is our baseline. Since the amount of outlier data are in the orders of magnitude more than normal data during testing, it is desirable not only to perform outlier removal, but also detect them using as little computational power as possible. Hence, the conventional random forest is modified to a ternary tree, where each node, apart from the original children, contains an additional ‘background removal’ node. During testing, when query data are propagated down the decision trees, outliers will be detected and rejected when they fall into a ‘background’ node before reaching the leaf nodes. Furthermore, when under low signal-to-noise ratio, the error accumulation in random forest is usually the cause of low performance, especially with high dimensional data as in our case. To reduce the amount of error, we propose using a ‘fuzzy split’ scheme, where, at each split node, ambiguous data near the decision boundary will be assigned to both left and right child nodes. Hence, the query template is more likely to hit the true positive leaf node. The results show that our approach outperforms the state-of-the-art approaches with regard to efficiency, while maintaining a comparable accuracy.

The contributions for this chapter are summarised as below:

- Proposed a novel randomised decision ternary tree for real-time 3D object pose estimation. Each split node in the tree carries an extra ‘rejector’ child for early outlier termination.
- The ternary tree is trained on LineMOD templates [Hinterstoisser et al., 2012a], a novel ‘fuzzy rule’ applied to the split function to deal with insufficient training samples.
- A split function for learning parameters of the ‘rejector’ node is proposed.
- A fast breadth-first leaf validation scheme is adopted for further speed-up.

Chapter 5. A Grouped-Outlier-Aware Registration-based method for Robust Ellipse Fitting

This chapter presents a case study of new outlier removal method which is capable of fitting ellipse in real-time under high outlier rate in a special case, based on the phenomenon that outliers generated by ellipse edge point detector are likely to appear as groups due to real-world nuisances, such as under partial occlusion or illumination change from shadows. To confront the grouped outliers while maintaining the fitting efficiency, we introduce a proximity-based ‘split and merge’ approach to cluster the edge points into subsets, following by a breath-first outlier removal process. The experiment shows that our algorithm achieves realtime performance under a wide range of inlier ratio and noise level with various types of realistic nuisances.

The contributions for this chapter are summarised as below:

- Proposed a novel outlier removal method to deal with grouped outliers that commonly appear in real-world ellipse fitting task.
- Proposed a dataset that consists of real images of elliptic industrial mechanical parts and synthetic elliptic shapes with noise and outliers.
- Applied the outlier removal method to speed-up and eliminate grouped outliers in an ellipse fitting method.

CHAPTER

2

A REVIEW OF OUTLIER REMOVAL METHODS

CONTENTS

2.1 Motivation	11
2.2 What is outlier	13
2.3 Challenges	14
2.4 Related works	16
2.5 Aspects of Outlier Removal Method	18

2.1 Motivation

Outlier removal methods are classical techniques that widely applied in pattern recognition and machine learning tasks. In the early time around 1970s, machine learning

methods, such as k-means clustering [MacQueen et al., 1967] or Gaussian fit, are widely applied for their simpleness but not robust enough to outliers. The outlier removal methods are naturally engaged to solve the problem. As time passed, sophisticated algorithms like Support Vector Machines (SVM) [Cortes and Vapnik, 1995] or Random Forest (RF) [Breiman, 2001] are supported to achieve better performance and also able to confront the noises or outliers in larger-scaled datasets. It seems that the outlier removal methods were no longer required to train a good classification model.

However, the demand of outlier removal methods are returned to life when the progress of the ‘Big Data revolution’ started in 2010s. For whatever the industry connected with ultra-fast broadbands, the incoming data are growing exponentially and everything seems digitalised. The state-of-the-art algorithms like deep neural networks (DNN) allow us to train very complex models with million real world data to build commercial-ready applications. At the same time, it is almost impossible to completely rely on human effort to clean and label the data, they are instead collected automatically in an unsupervised manner and filtered by outlier removal methods.

Apart from data mining, data are usually too redundant to be used effectively, especially in real-time tasks. When computer is asked to query an object of interest from a video clip, it may scan thousands of image frames. With a complex computer vision algorithm, this greatly wastes computational power and shall be preprocessed first by suitable outlier removal methods. Outlier removal can coarsely reduce search space in many machine learning systems to overcome high computational cost. This step is especially crucial when the signal-to-noise ratio is extremely low, such as in many real-world detection tasks. The region-of-interest sometimes occupy only a tiny portion of the whole image. In this case, applying the traditional sliding window to check all possible scale and location could be time consuming. While with a proper outlier removal technique, the ‘outliers’, such as proposals lie on the background, can be removed in a manner of lower order of complexity.

2.2 What is outlier

Outliers are considerably different from training data, or observations that deviate so much from other observations as to arouse suspicion that it was generated by a different mechanism [Hawkins, 1980]. As the toy examples shown in Fig. 2.1, a datum is considered outlier if it is sufficiently distinct from the majorities. For example in the figure (b), most data points $(x_i, y_i), i = 1, \dots, n$ roughly follow a line function where $y = \alpha + \beta x$, except an outlier data point that does not fit at all.

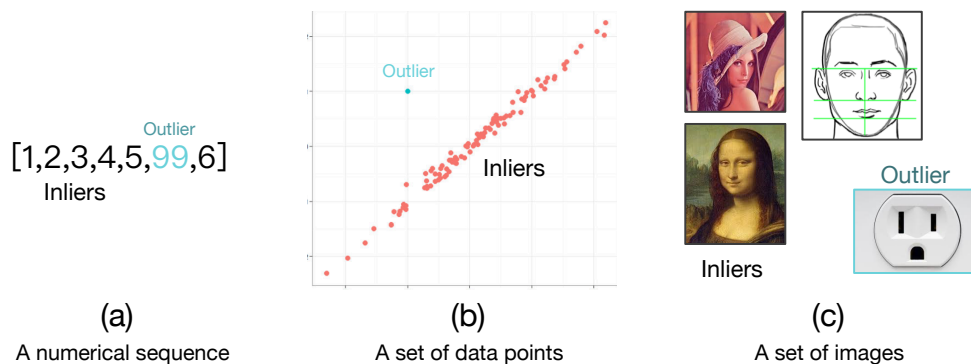


Figure 2.1: Toy examples of different outlier types.

Outliers occur with various reasons, such as entry errors from human or unusual behaviours in the target subject, but the term should be distinct from the 'noise', as well as they are dealt with different techniques in general. An outlier can be meaningful and generated from a 'true' signal, while noises are mostly void and should be fixed or discarded. Apart from removing outliers, sometimes outliers are more of interest to the data analyst, these methods are widely applied in anomaly detection, for surveilling terrorist or malicious activities. Moreover, same techniques are used for novelty detection to learn new patterns from the data, such as new trends in social media.

2.3 Challenges

A natural approach for outlier removal is to define decision boundaries around ‘normal observations’ so that the rests are declared as outlier. However, there are many factors that make it very challenging for real-world tasks.

First, precisely locating such boundaries can be extremely difficult, especially when there is little prior knowledge of the inlier distribution. That means an observation that near to the boundary is likely to be misclassified. Even worse, the decision boundary may change and evolve dynamically over time.

Second, even within the same pieces of data, there is no universal definition of what an inlier or ‘normal observation’ is and what is not, since it is always binding to the task. For example, the region-of-interests in a positive dataset for object recognition, detection and semantic segmentation tasks can be quite different, as illustrated in Fig. 2.2). This makes the setting of inlier boundaries task specific, hence imply different outlier removal approach.

Third, despite outlier and noise are generated by different mechanic, they may appear to be the same. When both related techniques are applied, they might inference with each other and produce suboptimal results. For example, if we misclassify a part of noises as outlier and removed first, the estimated noise model would be biased then the quality of de-noising is decreased, and vice versa.

Fourth, labelled data is often still required to train a model used by outlier removal methods.

Finally, when deploying real-world applications with outlier removal as preprocessing stage, it should has much less computational cost than validating an inlier. Also, despite it may save an order of magnitude or more computations, it should avoid

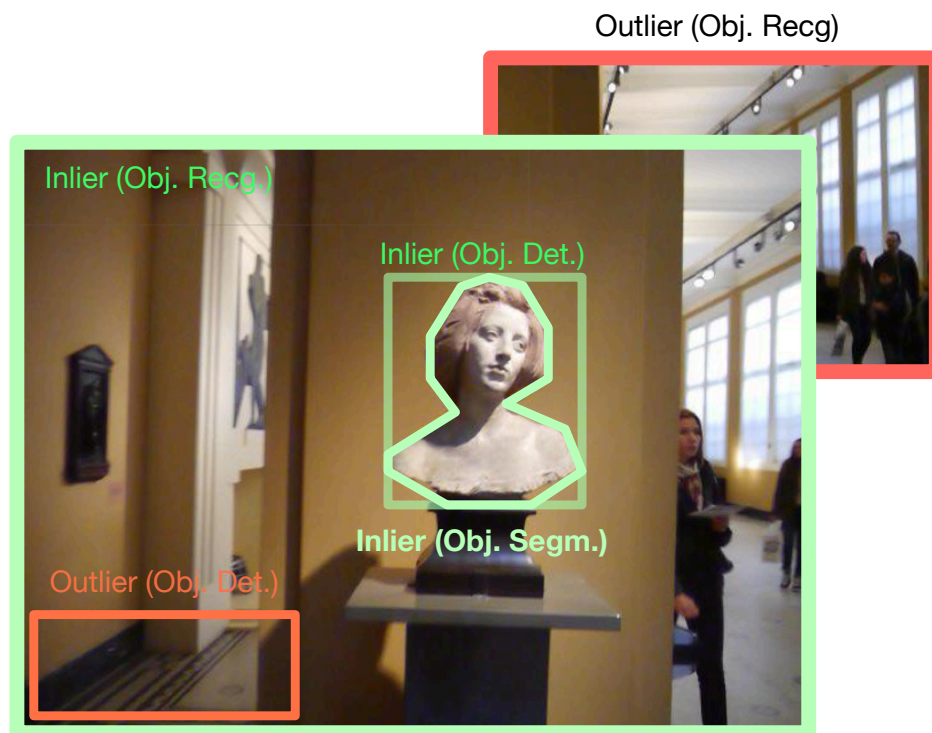


Figure 2.2: Different pixel-region-of-interest in a positive dataset for object recognition, detection, semantic segmentation task.

to affect the overall precision of the entire system. A trade-off between precision and recall needs careful decision.

2.4 Related works

As illustrated in Fig. 2.3, vast amount of applications can be benefitted from outlier removal methods, with not less technical approaches.

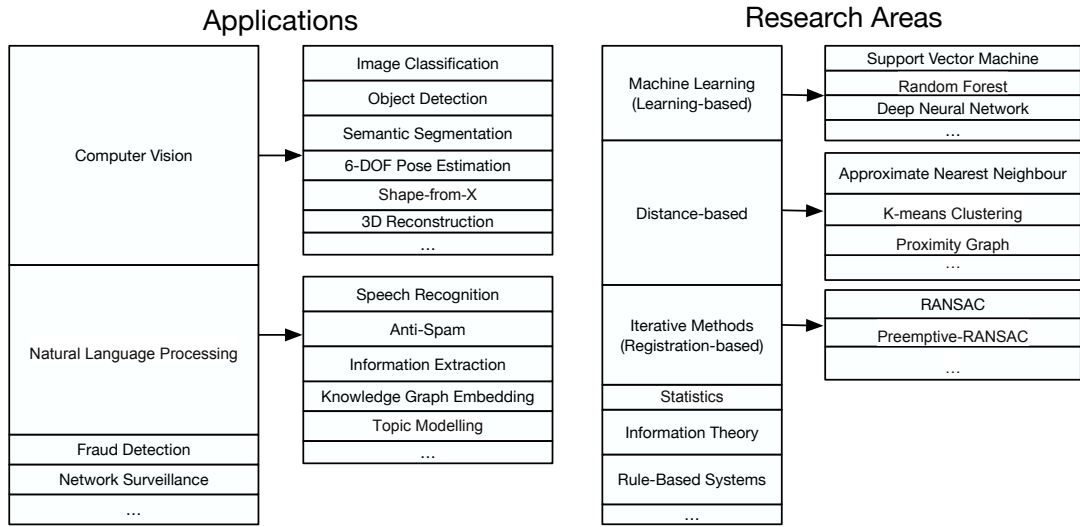


Figure 2.3: Applications that utilise outlier removal methods and the research areas behind the techniques.

In this thesis, we focus on computer vision applications with aids of outlier removal methods.

2.4.1 Computer Vision Applications

For computer vision tasks, as illustrated in Fig. 2.4, as long as there exists uninterested observations or irrelevant data, whether in training or testing stage, it usually worths the effort to find an appropriate outlier removal method to boost the overall performance and speed.

For image classification problem, a popular large-scale face recognition dataset

FaceScrub [Ng and Winkler, 2014a] drew celebrity public figures from search engine, then applied outlier detection classification and gender classification to validate visual similarity for automatic data cleanse. A recent work on face recognition also show that a refined dataset provide much better classification performance [Deng et al., 2018] (a claim from the discussion in the author’s Github issue).

Some object recognition methods heavily rely on feature matching, taking the feature descriptors of interest in a set, then match all other features in another set using some distance calculation. One straight forward example would be calculating the fundamental matrix between two camera views in the same scene, by finding matching local features in both images. This step is crucial in many industrial applications, including 3D reconstruction, motion tracking, object recognition, robot navigation and more. There are lots of works on finding the interest points and constructing invariant local descriptors, such as the classical SIFT descriptor [Lowe, 2004a] or a more recent work KAZE descriptor [Alcantarilla et al., 2012]. Here we assume the images are processed into a set of some local feature descriptors.

LineMOD [Hinterstoisser et al., 2012a] applies two simple but effective outlier removal steps based on depth and colour thresholding check. This results a great reduction in computational complexity and thus the overall pipeline achieves state-of-the-art speed at the moment. For recent works with convolutional neural network, MTCNN [Zhang et al., 2016] proposed a deep cascaded multitask framework to predict face and landmark location in a coarse-to-fine manner; Faster R-CNN [Ren et al., 2017] introduced a Region Proposal Network (RPN) for fast generating detection proposals and then fine-tuning for object detection.

For yet another example related to pair-matching between image local descriptors, such as in a 6-DOF pose estimation, SLAM (simultaneous localisation and mapping) or SfM (Structure-from-Motion) system, outlier removal methods are applied to quickly discard most of the mismatched correspondences, or ‘negative’ descriptor candidates

that do not follow the geometric or spatiotemporal constraints.

In a curve fitting problem, edge curve and points are extracted in order to perform an industrial shape fitting task. Outliers can arise from the contaminations on the object, the sensor or inaccurate edge extraction method. As the generation mechanic of ‘outliers’ can be completely different from the ‘noise’ that we expects, outlier removal method should be applied as a pre-process stage to ensure the performance of following shape fitting algorithm.

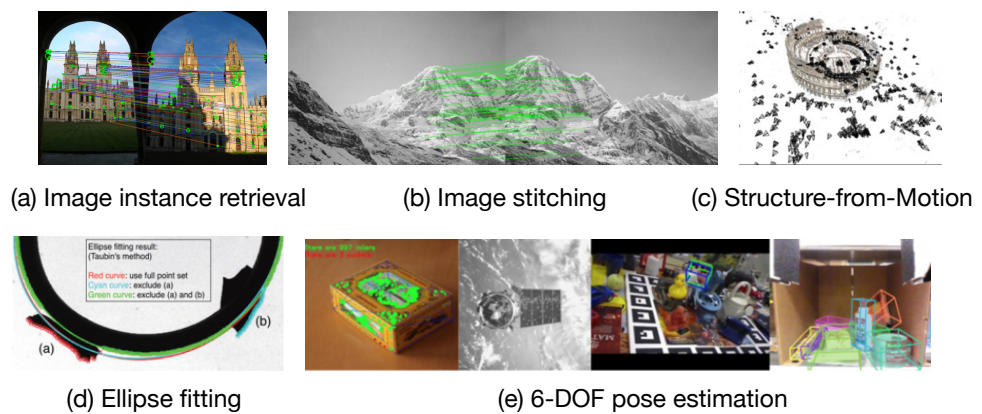


Figure 2.4: Computer vision applications that heavily utilise outlier removal.

2.5 Aspects of Outlier Removal Method

In this subsection, we split outlier removal methods into four factors: data type, data relation type and label. All factors greatly affect the decision of which approach we should choose to solve the problem.

2.5.1 Data

The first important aspect of all outlier removal methods would be the nature of input data, i.e. the inner-relationship of data. In the book of *Introduction to data mining* [Tan, 2018], data are a collection of data objects: they can be records, points, cases, samples or entities. Each data instance can be further described into attributes, also known as variables, fields, characteristics, dimensions, or features. An attribute can be discrete or continuous. Discrete attribute has only a finite or countably infinite set of values, such as a set of word or zip codes, binary value is included as a special case. Continuous attribute has real numbers as attribute values, such as pixel intensity or location, they are often represented as floating point variables. Furthermore, a data instance usually consists of multiple attributes, i.e. multivariate, and can be a mixtures of attribute types.

The nature of data attributes determine the availability of outlier removal methods. For methods based on machine learning or statistical models, the data are required to be continuous or categorical values. While when pairwise distances between data are more of interest, distance-based methods are suitable, e.g. approximate nearest neighbour or proximity graph.

In other hand, important information exist in the relations between data, sometimes we treat the set of data as a whole. For example, a video sequence contains temporal information (timestamp) and spatial information (image pixel location), and these information are vital in many applications such as behaviour analysis. In 3D reconstruction or 6-dof object pose estimation problems, we also assume the rigid objects share consistent geometric relationship across different point of views. In such cases, we might treat the data as a graph or a set of graphs.

2.5.2 Relation

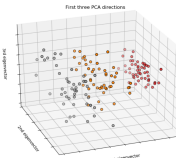
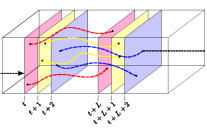
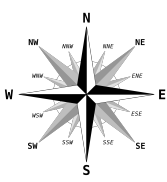
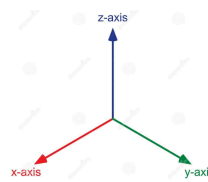
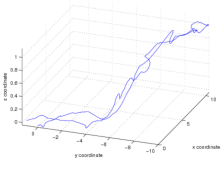
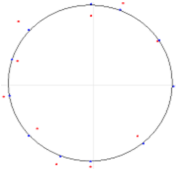
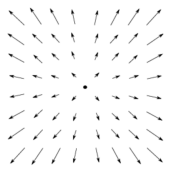
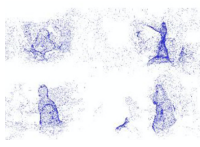

	Discrete	Continuous	Sequential/Graph
Individual	<p>A A A A A A A A A B A A A A A A A A A A</p> <p>(1a) A letter B in a set of A</p>	 <p>(1b) A data point in a feature space</p>	 <p>(1c) A trajectory in feature space</p>
Contextual	 <p>(2a) A cardinal direction</p>	 <p>(2b) A 3D location</p>	 <p>(2c) A trajectory in 3D geometric space</p>
Behavioural	<p>A B C D E 1 ○ ○ ○ ○ ○ 2 ○ ○ ○ ○ ○ 3 ○ ○ ○ ○ ○ 4 ○ ○ ○ ○ ○</p> <p>(3a) An answer in answer sheet</p>	 <p>(3b) A edge point of a circle</p>	 <p>(3c) A visual trajectory from egocentric video</p>
Collective	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p> <p>(4a) Human language</p>	 <p>(4b) 3D point clouds</p>	 <p>(4c) Trajectories of location of crowds</p>

Figure 2.5: Examples of data type versus relation type.

2.5.2.1 Individual

In this subsection, we further categorise the data by their inter-relationship, which is also deterministic in choosing an appropriate outlier removal method. There are four relation categories of data attributes: individual, contextual, behaviour and collective. In Fig. 2.5, we present a table with examples for each combination of input type and relation type.

For the simplest relation ‘individual’, data is treated as a set of individual instance, i.e. we do not have to query other instances to distinguish an outlier data. Most types of data used for outlier removal method fall into this category.

In the example of Fig. 2.5 1a, without knowing other data, we can safely classify a letter as outlier as long as it is not an ‘A’. In machine learning problems, as illustrated in Fig. 2.5 1b, we often represent data as high-dimensional vectors in a feature space, so that we can easily split or measure the data similarity. Similar techniques can be applied to remove the outlier data point. In Fig. 2.5 1b, a trajectory, i.e. a data sequence or graph, in feature space may represent a ‘special move’ of a data point, it has uses in motion recognition or some surveillance systems.

Learning-based methods are often adopted to approach this problem, by training a classifier (SVM, boosting, decision forest, neural network, etc.) to classify outlier data. Since only ‘normal’ data are available during training, this is also known as ‘one-class learning’. The essence of learning-based methods is to find a tight decision boundary based on the distribution of normal data. A common example is the series work of one-class SVM [Schölkopf et al., 1999, Tax and Duin, 1999], as illustrated in Figure Fig. 2.6. Unlike conventional two-class SVMs, one-class SVM [Schölkopf et al., 1999] determines the novelty boundary in the feature space corresponding to a kernel by separating the transformed training data near the boundary, i.e. , support vectors, from their origin in the feature space with maximum margin.

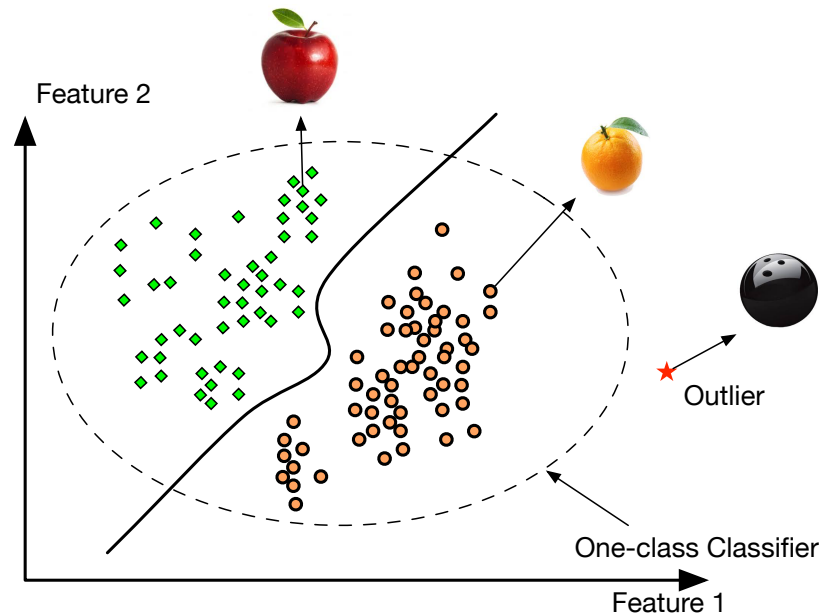


Figure 2.6: A one-class classifier applied to a toy example. The solid line represents the conventional classifier that distinguishes between ‘apple’ and ‘orange’, while dash line is the one-class classifier that detects the outlier that does not belong to any of classes.

[Tax and Duin, 2004] proposed another support-vector-based method called the support vector data description (SVDD). Instead of hyperplane, SVDD uses the minimum hypersphere that contains most normal data as the outlier boundary. The method is illustrated in Figure Fig. 2.7. The method is claimed to be robust against outliers in the training set and is capable of tightening the description by using negative examples.

Another line of work is to expand one-class learning into ensemble learning. Inspired by the barrier methods from the theory of constrained optimisation, [Rätsch et al., 2002] proposed one-class boosting, which uses a convex combination of base hypotheses (decision stumps) to decide whether or not a query point is an outlier. [Désir et al., 2013] designed one-class random forest by synthesising outlier samples in the places where training data points are sparsely located.

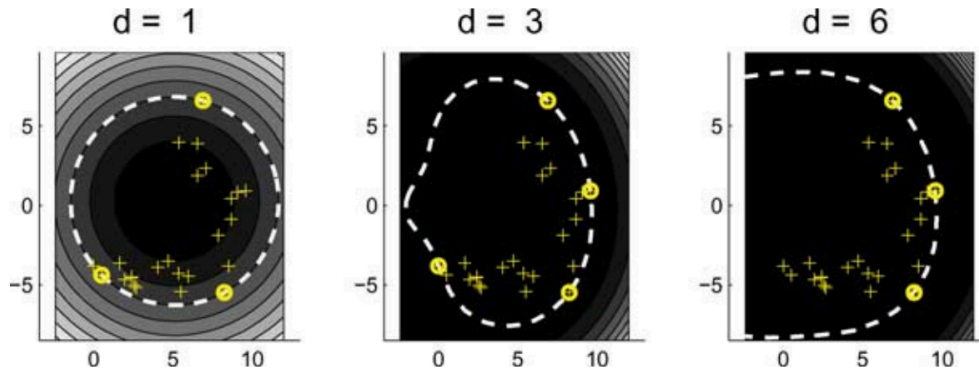


Figure 2.7: Figure 2 in the paper of [Tax and Duin, 2004]. It shows a tightly trained data description on a banana shaped data set, a good description should cover all target data but includes no superfluous space. Outlier removal problem can make good of this boundary. They also applied a polynomial kernel with varying degrees. Support vectors are indicated by the solid circles, the dashed line is the description boundary.

2.5.2.2 Contextual

Some data need context to be meaningful, location is one of the typical examples. Depending on the starting point, different combination of directions may point to the same place while same directions may result in different places. So whether a contextual data is an outlier depends on its relationship with other data but not its 'absolute value'. Time-series data is also often treated as contextual data.

When the data are contextual, the most straightforward approach is leveraging the neighbourhood. However, a real world scene usually captures a large number of distinctive features, thus pairwise matching through exhaustive search is too expensive to be adopted. The non-parametric nature usually takes more computation than we can afford To check if m data points are outliers in a dataset containing n data points, the complexity of determining its k -nearest neighbour distance is $O(nm)$. Hence it is obviously infeasible to be directly applied to large datasets .

The widely used methods to accelerate the process are known as approximate near-

est neighbour (ANN) methods, aiming to measure the distance from the query to a large set in sublinear complexity, often in a high dimensional feature space. One choice is the k -nearest neighbour graph (k -NN) approach by [Ismo et al., 2004], where data points that have large k -nearest-neighbour distance are considered as outliers. A simplified example is illustrated in Figure Fig. 2.8. A large number of literatures has discussed about improving the accuracy or speeding up NN methods [Muja and Lowe, 2014, Behmo et al., 2010, Boiman et al., 2008, McCann and Lowe, 2012, Tuytelaars et al., 2011]. However, these methods only care about the matching performance on a single query feature and every match is assumed to be valid or of interest. In many cases, most of the queries are completely irrelevant, i.e. outliers, causing waste of computational power and sometimes this even disturbs the results of later stages.

To deal with irrelevant queries, or the ‘outlier’ queries, distance-based outlier removal methods are adopted. One of the merits of the distance-based outlier removal approach is its ability to distinguish between inlier with additional ambient noise and isolated outliers. This is because ambient noise usually causes lower k -nearest neighbour distance but similar point-to-cluster-distance after clustering is applied.

There are various methods to reduce the complexity. The common solution is two-fold: through index structures or pruning tricks. Index structures improve the speed of data retrieval operations, but effectiveness degrades if the data is high-dimensional; pruning trick or ‘early termination trick’ ignores data that have upper-bound estimate on the k -nearest neighbour distance value below the r th best outlier score found so far. A more concrete review of both approaches is presented in [Aggarwal, 2015].

In the literature, [Zhang and Wang, 2006] propose a High-Dimension Outlying Subspace Detection (HighDOD) method with a search algorithm dynamically deciding which subspace to search or prune. The decision is made based on a score that is computed with global and local threshold pruning strategies. Compared to the original k -NN, this method works efficiently in high-dimensional space. [Angiulli and Fassetti,

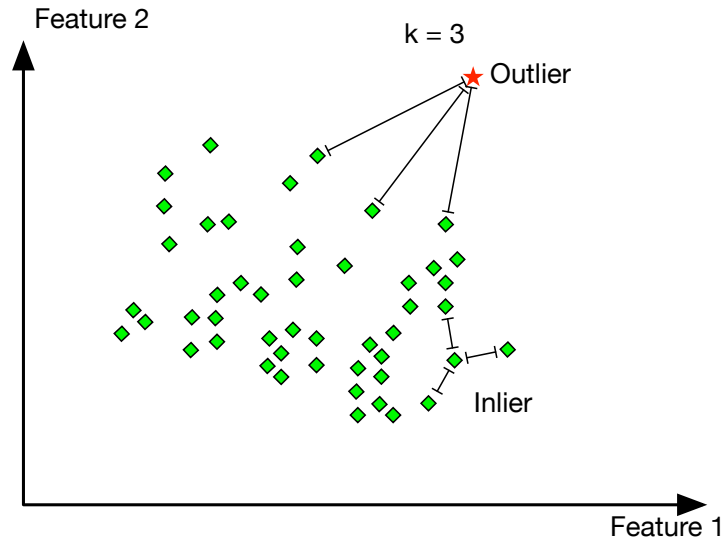


Figure 2.8: A k -nearest neighbour algorithm applied to a toy example, where $k = 3$. An outlier is detected when the sum of distance of its nearest neighbours is significantly larger than other data points.

2007] address detecting outliers in streams of data that have limited memory requirements, and returns an approximate answer based on accurate estimations with a statistical guarantee. [Ghoting et al., 2006] present a method called Recursive Binning and Re-Projection (RBRP), a two-phase log-linear algorithm, for mining distance-based outliers, particularly targeted at high-dimensional data sets.

In this thesis on Chapter 4, we adopt the idea from learning-based outlier removal methods to learn the one-class ‘rejector’ split node in the randomised ternary tree. The basic idea is to extract a representative subset of features, then ‘early reject’ the negative sliding window proposals by a newly designed split function in randomised decision ternary tree, so that the computation from fully validation is greatly saved. This parametric approach remove the outliers in a coarse-to-fine manner and has logarithmic computational complexity that can deal with dataset that contains large number of

object classes. The final validation stage adopts the distance-based method to avoid overfitting due to insufficient but high-dimensional template features in each object class. The preview is illustrated in Fig. 2.9. This approach can be potentially adopted to any system that has large number of negative proposals.

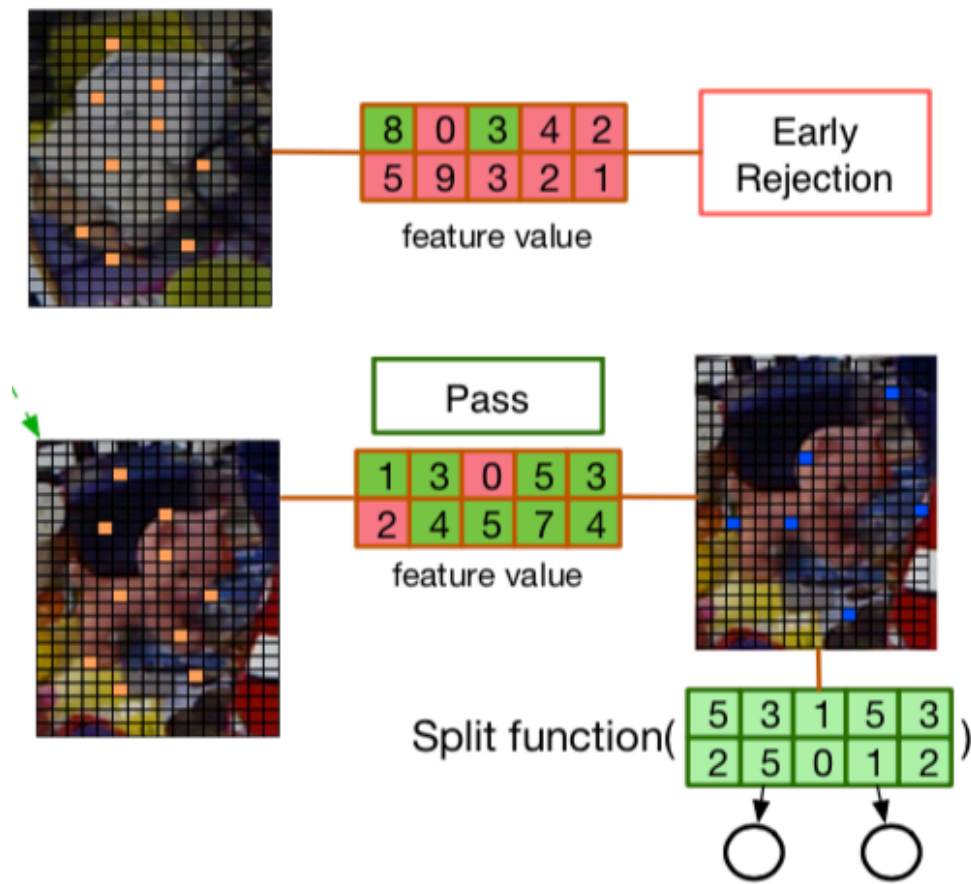


Figure 2.9: Early rejection of background queries in Chapter 4.

2.5.2.3 Behavioural

Data are behavioural if it tends to record behaviours under different conditions. One example shown in Fig. 2.5 3a is an answer sheet students use in the exam. The same answer can be either correct or not, depending on its question index. The points in Fig. 2.5 3b are inlier given a set of circle parameter, however they are outlier if the circle parameter changes. In 3c, a visual trajectory formed in egocentric video depends on the egocentric motion.

For outlier removal problems involving behavioural data, if we have the knowledge of how conditions and data are related, the whole problem can be deducted into subsets for each of the conditions, so that we can treat behavioural data as individual data to apply simpler techniques to solve the problem.

However, in most real-world applications, the relationship between conditions and data are missing. Or even worse, the conditions are unknown, the only information we have is that the data are behavioural and there are a known type of inter-relationship within them. In some cases even the condition is completely latent.

In computer vision, the corresponding problem to this outlier type is geometric fitting of shapes or visual fixation on objects. There are further problem settings, one is to find the condition that most fit with the behavioural data. For example, given a set of points, including outliers, then find the shape parameters that can maximise the number of pairwise matches. Another problem setting is the opposite: find the subsets of behavioural data that match with a given condition. For example, given one or a set of fixed shape parameters, find out if the point sets generated by candidates exist in the full point set. The first is often seen in geometric fitting problem, the latter is seen in object image registration or pose estimation problem.

Solutions for both problem setting involve repeatedly estimating the transformation between point sets, where registration-based outlier removal method plays an im-

portant role. The challenge is that ambiguous, irrelevant data or false matches are very common in such problems, mostly much more than the positive matches.

To achieve the goal with limited computation resource, we need to first reduce the number of pairwise testings between conditions and data, then validate all pairs to find the inlier data subsets and corresponding conditions. The cost depends on the complexity of transformation and its technique behind. These techniques are generally involved in registration-based outlier removal method.

Registration-based methods, also known as model-fitting methods, are done by iteratively fitting a model to a set of query data that contains outlier points. When completed, all data points far from the model are considered outliers. One classic example is random sample consensus, or RANSAC in short, proposed by [Fischler and Bolles, 1981a], illustrated in Fig. 2.10.

Since outliers are generated from other sources, which are completely irrelevant to the true source, they should be removed first before calculating the model parameters. Therefore, registration-based methods, e.g. , RANSAC-like algorithms, usually operate in a hypothesise-and-verify framework: a minimal subset of the input data points is randomly selected and model parameters are estimated from this subset [Raguram et al., 2008]. One advantage of this approach is that it can accurately estimate the model parameters under a very low outlier ratio as long as the computation time allows. However, as a non-deterministic algorithm, it takes no upper bound on the iterations to complete the computation. In other words, more iterations increases the probability of producing a better result. The number of samples required also exponentially increases with the outlier level.

Registration-based methods for outlier removal have a long history. RANSAC Fig. 2.10 was proposed in 1981, and became a fundamental tool in many computer vision applications. Later works were shown to improve the effi-

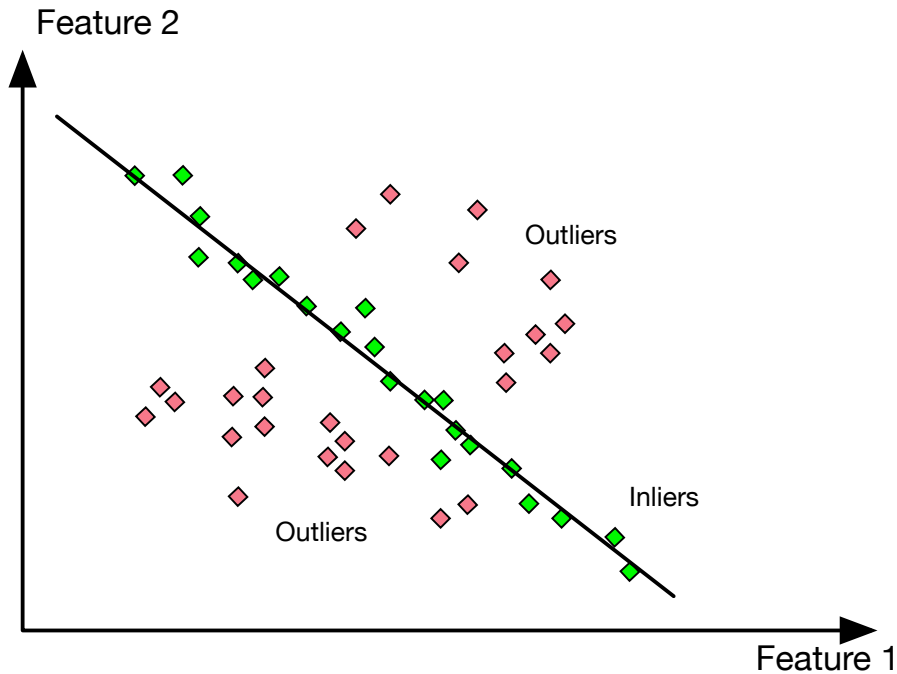


Figure 2.10: A simple example of registration-based outlier removal method (RANSAC). Assuming the point set contains inlier that can be fit to a line, this approach iteratively finds the best line that fits the most data points. All the data points that do not fit the given model are considered outliers.

ciency of the algorithms and the accuracy of the estimated parameters, and reduce the dependency on the setting of problem-specific constants. [Chum and Matas, 2002] improved efficiency by introducing a pre-evaluation stage to quickly filter out the bad hypothesis. [Torr and Zisserman, 2000] presented a robust estimator, MLESAC, which is a generalisation of the RANSAC estimator. It evaluates the quality of the consensus set, i.e., the data randomly sampled for parameter estimation, to provide a good initial estimation. [Nistér, 2005a] proposed a practical breadth-first preemption scheme to accelerate the RANSAC process when the amount of candidates is fixed.

In this thesis on Chapter 3, we study the use of outlier removal methods from var-

ious aspect and adopt a registration-based method to reconstruct 3D mesh models during training, then fit the model to query feature points to decide target location and pose whilst removing outliers. By combining with another spatial-temporal constraint, most of false matched local descriptor pairs are eliminated and thus the best performance is achieved.

2.5.2.4 Collective

Collective relationship is relatively complicated. It usually needs the whole dataset to determine if a data instance or a data subset is outlier, i.e. a single outlier or a set of outliers might appear to be inlier themselves. In the example of Fig. 2.5 4a, human language is complex and it is often hard to comprehend the meaning of a sentence until read the whole paragraph. Fig. 2.5 4b and 4c show other examples when data is continuous or represented as graphs. In applications relate to anti-cyberattack or security surveillance, unusual behaviours are also a sequence of events or actions, collective outlier (also referred as anomaly) has to be explored to solve the problem.

In this thesis on Chapter 5, we propose a small improvement based on the classic RANSAC to deal with outliers that locationally grouped. Despite this assumption conflicts the general definition of outliers, which an outlier cannot be modelled or predicted, we found that our approach can improve the shape fitting performance in real world industrial tasks. In this approach, we apply a ‘split and merge’ technique to group the edge points into subsets, i.e. collections of edge points and estimated ellipse parameters.

2.5.3 Label

A label denotes whether the data instance is an inlier or outlier. In most machine learning tasks, labels are obtained from a reliable source, i.e. a human or other pre-

cise measurement tools, they require lots of effort and can be extremely expensive for large-scale. Different from general classification problem, outlier is a kind of special class that may have any patterns. In other words, it is usually an 'A or not A' problem instead of 'A or B'. Moreover, since outlier is defined as 'a true signal that does not follow the normal pattern', the occurrence of an outlier is much rarer than an inlier, thus collecting a set of outliers is also difficult.

That said, despite outliers are generate from different mechanics from inliers, in reality it is worth trying to learn the mechanics that causes unwanted data. In Chapter 5, a case-study of outlier removal in ellipse fitting with 'grouped outliers', we have found that a great portion of outlier edge points are generated from the edge of other objects or due to deformation of target objects. The observation is that in industrial shape fitting problems, outliers appear to be in group in almost all the cases. Eliminating particularly the grouped outliers has better performance than general outlier removal method. This is a case that labelling outliers may also improve the outlier removal method.

2.5.4 Evaluation Metrics

To solely measure how well an outlier removal method performs, we often considered it as a two-class classification problem, i.e. classifying the inlier and outlier. Then, various standard evaluation metrics can be applied, e.g. accuracy, receiver operating characteristic curve (ROC-curve), precision, recall, and etc.. This approach is mainly used to evaluate the scalability, ability of handling noise of outlier removal methods. However, due to the need of ground truth, the evaluations are usually based on synthetic datasets, or data with artificially injected outliers, such as several benchmark datasets in UCI Knowledge Discovery in Databases Archive (UCI KDD) by [Bay et al., 2000].

In most machine learning applications that based on real world data, labelling inliers or outliers is infeasible. For example, the number of detected local interest points on each image is typical a hundred to thousands in average. As the works in this thesis are application domain oriented, outlier removal methods are evaluated indirectly from the application performance. Since in most applications, outlier removal method is implemented as an extra stage to the original workflow, it is important to show the overall improvement of either or both efficiency or accuracy.

SIFT Matching
SIFT Matching + RANSAC
Bag-of-Words
Video Google
ScSPM
NBNN
NBNN + RANSAC
Local-NBNN
Local-NBNN + RANSAC
ScSPM + KPA
Video Google + KPA
Local-NBNN + TM
Local-NBNN + TM w/o averaging
Local-NBNN + TM + KPA
2D-to-3D Matching + ePnP
2D-to-Trajectory Matching + ePnP
3D-to-3D Matching + 3D RANSAC

Figure 2.11: Experiment list for methods with/without outlier removal approaches in Chapter 2.

In this work, the effectiveness of several outlier removal approaches are evaluated by measuring accuracy or presenting a precision-recall curves with and without the additional outlier removal process. For instance in evaluation section 3.5 of Chapter 3, several video-based object recognition methods are evaluated, the experiment list is shown in table Fig. 2.11. For execution time of each method, the time consumption of processing one video frame of each workflow is given. Despite the time is coarsely presented due to the large diversity in implementation environment, for which methods are suitable to real-time application is clear. For 6-dof object pose estimation in Chap-

ter 4 and ellipse fitting in Chapter 5, a more detailed comparison between baselines and proposed method is presented, including time and average accuracy through all object classes.

3

CHAPTER

OUTLIER REMOVAL IN VIDEO-BASED OBJECT RECOGNITION: A COMPARATIVE STUDY

CONTENTS

3.1 Overview	36
3.2 Approaches	38
3.3 Implementation	44
3.4 Dataset	48
3.5 Evaluation	52
3.6 Summary	60

3.1 Overview

As mentioned in Chapter 1, a challenge of the object recognition problem is accommodating numerous nuisances. To achieve this, one needs to collect and annotate enough image samples to cover different scenarios during training, which is tedious. A more cost-effective solution is to use videos, which tend to yield a more complete description of their content than individual images, and require only sequence-level labels. Moreover, all these nuisances also lead to a significant portion of outlier feature points during testing, which not only confuse the classifier, but also bring down the run-time speed. Thus, to be able to detect and remove these features motivates the use of outlier removal methods in this problem.

From videos, one can exploit temporal information and the underlying 3D spatial structure of the target object (through 3D reconstruction). A recent research [Li and DiCarlo, 2010] reveals how the human brain can effortlessly interpret a multitude of objects with different identity-preserving transformations. After exposing a monkey’s visual system to an artificial visual world without temporal contiguity, neuroscientists observed that inferior temporal cortex neurons began to lose their capacity for being transformation invariant. This strongly encourages the exploitation of temporal information in object recognition tasks.

On the other hand, the exploitation of spatial cues to identify foreground and background (outlier) features, either in 2D image layouts [Lazebnik et al., 2006] or 3D object structures [Gordon and Lowe, 2006b], is a flourishing branch of object recognition. The viewpoint-invariant theorem [Peterson and Rhodes, 2003] states that the essential component of object recognition, regardless of viewing conditions, is structural information. Encoding object structural information requires only a small amount of memory, yet it is capable of producing a multitude of object representations via their

interrelations and mental rotations. In the field of computer vision, stereo vision is often utilised to obtain precise depth perception, and hence 3D structure. On top of that, some recent studies have achieved impressive performance by using multi-view images to reconstruct 3D information to support object recognition [Knopp et al., 2010], semantic segmentation [Bao et al., 2012] and pose estimation tasks [Savarese and Fei-Fei, 2007, Gordon and Lowe, 2006b]. Recently, due to the growing use of wearable vision devices, e.g., *Google Glass*, research into egocentric videos has attracted more and more attention. As one of the useful source of spatial information, egocentric vision has the advantages of being controllable during capturing informative viewpoints and being more practical than turntable settings.

Yet, it is undeniable fact that using video as data source is likely to suffer from redundant information, such as near-duplicated frames. Video data also tends to contain more visual nuisances than still images, such as motion blur or occlusion. We believe that to be successful in video-based object recognition, it is necessary to compare different ‘object representations’ and their matching/outlier removal strategy to identify the most promising approach.

In this chapter, our goal is to investigate the use of spatial and temporal informations in videos in order to perform outlier removal, and hence, better video-based object recognition (VbOR) in realistic scenes. In particular, we aim to answer the following questions: are they helpful? If so, are they helpful in terms of accuracy or efficiency? Can they be combined? It is worth noting that there have been recent advances in object *category* recognition [Deng et al., 2009, Everingham et al., 2010, Lin et al., 2012], but only a small number of studies have investigated the problems of *instance* object recognition [Rothganger et al., 2006], particularly with video [Ren and Gu, 2010, Liu et al., 2014]. Therefore we highlight our contributions as below:

- We captured an instant-level object recognition dataset in video called *Sculptures*

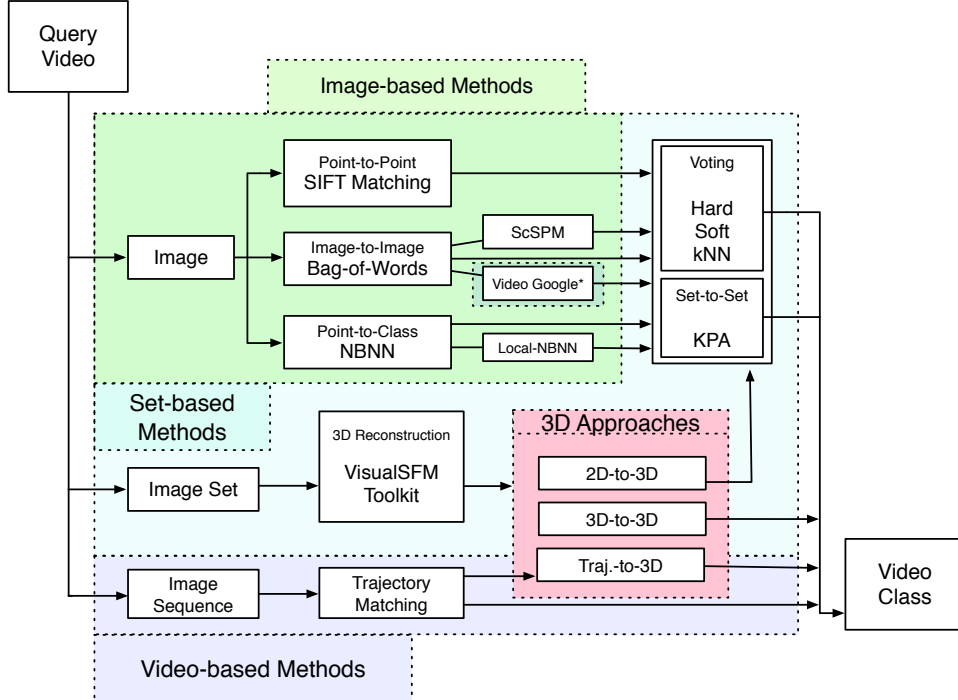


Figure 3.1: Method categorisation and experimental setup.

in Victoria and Albert (V&A) Museum Dataset from an egocentric viewpoint.

- We categorised and compared diverse state-of-the-art object recognition frameworks and their video-based extensions.
- We analysed how outlier removal methods improve the existed video-based object recognition systems, hence proposed a hybrid solution that combines the advantages of both temporal and spatial cues.

3.2 Approaches

Given exemplar videos of target objects, our purpose is to identify them in query videos. Due to the egocentric setting in our study, each video captured multiple views of only one target object that appeared roughly in the centre. Therefore, the whole

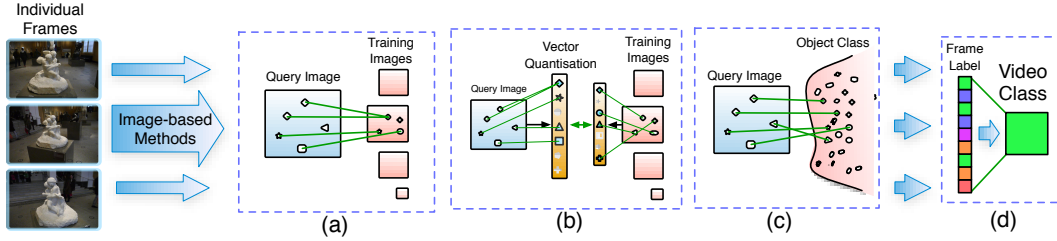


Figure 3.2: Image-based methods selected from three state-of-the-art object recognition frameworks.

video was assigned and recognised with one label. In this comparative study, we focused on the methods represented by the taxonomy shown in Figure 3.1. In terms of utilising spatial information, these methods can be categorised mainly into 2D and 3D approaches. Among the 2D approaches, there are three different ways to represent videos: image-based, set-based and video-based. In image-based methods, each video is treated as independent images, where a straightforward combination of individual results is applied to obtain the final output of the video. In set-based methods, each video is treated as a set of unordered images with underlying mathematical structure, such as a manifold. In video-based approaches, each video is represented as a set of ordered images, i.e., with temporal information. By contrast, 3D-based VbOR utilises reconstructed 3D information from multi-view images. This is a relatively new area with only a small set of methods. Thus we consider these method as a separate category.

In the following subsections, we analyse the pros and cons of each framework. Comparative evaluation can be found in Section 3.5.

3.2.1 Image-based Methods

To select representative image-based methods, we adopted three baselines from state-of-the-art object recognition frameworks based on their image classification

techniques: (a) point-to-point (P2P), (b) image-to-image (I2I) and (c) point-to-class (P2C), as illustrated in Figure 3.2. The image classification results are combined later via voting.

Point-to-point methods measure the similarity between two images based on their corresponding local image appearance, which is usually encoded by a feature descriptor.

In the seminal paper by Lowe [Lowe, 2004a], image classification was performed by matching a set of keypoints detected in image regions. Using robust fitting algorithms, e.g., *RANSAC* [Fischler and Bolles, 1981b], the correspondences can be constrained further by dominant transformation between the matched pairs. This technique can improve recognition precision significantly. However, it may fail when there is no similar viewpoint in the database to a query image. Recent advances in graph matching [Cho and Lee, 2012, Duchenne et al., 2011] have relaxed the geometric constraint between point correspondences for articulated or deformable object recognition. However, these methods are generally computationally expensive and infeasible for large-scale problems.

Image-to-image methods compute the vector of visual word frequencies in images to facilitate similarity measurement. In general, I2I methods are efficient and suitable for large-scale problems because of the compactness of their image representations. The Euclidean distance in a feature space reflects the similarity between features. Thus we can also apply learning-based classifiers, e.g., linear support vector machine (SVM) and Random Forests, to facilitate better generalisability and efficient recognition. I2I methods have been applied widely to various image classification tasks, e.g., scene recognition [Lazebnik et al., 2006], image categorisation [Yang et al., 2009, Bucak et al., 2014], object recognition [Liu et al., 2011] and video image retrieval [Sivic and Zisserman, 2009]. These methods have achieved state-of-the-art performance

on most publicly available benchmark datasets of image classification. [Deng et al., 2009, Everingham et al., 2010]. However, despite the success of these methods, the vector quantisation process may degrade the discriminatory power of individual image features, which is crucial for instance recognition problems.

Point-to-class (P2C) methods have also achieved impressive results on several benchmark datasets in recent years. The concept was emphasised in [Boiman et al., 2008] to sidestep the negative effects of vector quantisation in I2I methods, and later improved and extended in [Tuytelaars et al., 2011, Behmo et al., 2010]. The basic idea is to directly measure the similarity between query features and training features in every object class without vector quantisation. Compared to P2P methods, P2C has better generalisability, because images are decomposed into image features that can be matched simultaneously across all training images. This approach is also suitable for large-scale problems because the feature-matching procedure can be accelerated to real-time using approximated nearest neighbour algorithms. The main drawback is that P2C methods are based on non-parametric classifiers and consequently consume more memory because all the features are retained.

3.2.2 Set-based Methods

Set-based methods aim to capture the inherent characteristics of a set based on the assumption that the members of the set follow a particular statistical distribution, as shown in Figure 3.3. For the VbOR problem, the appearance of an object in each frame is constrained by identity-preserving image variations, i.e., viewpoint, scale or illumination changes. If we consider that an image is a data point in a high-dimensional space, the manifold spanned by the image variations can be learned using subspace or manifold techniques [Mei et al., 2011, Wolf and Sashua, 2003]. The video-to-video

similarity can then be estimated based on their manifold intersection, e.g., their largest principal angle. In previous studies, these techniques have achieved superior performance in different tasks, such as face recognition [Wang et al., 2012], head pose estimation [Wu and Trivedi, 2008] and object pose estimation [Mei et al., 2011].

The motivation for applying set-based techniques to object recognition problems is that the unseen views of an object can be interpolated from existing images, which leads to significant improvements in generalisability. However, in dynamic real-world scenarios, estimating the subspace or manifold from an image set is always challenging, as the distribution of images in a set is often highly non-linear due to the existence of complex background noises and object variations.

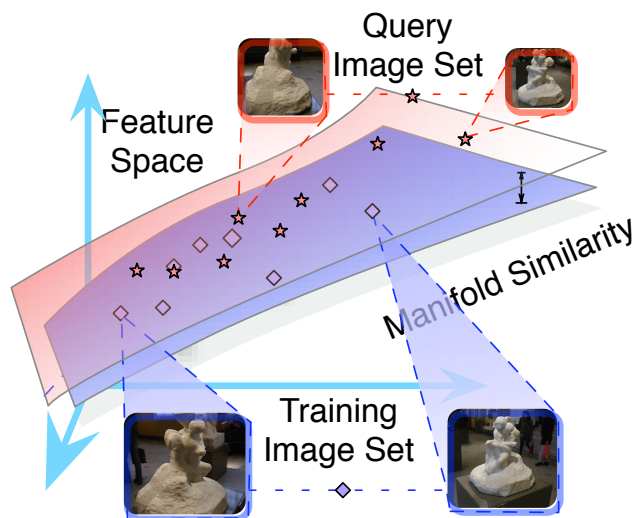


Figure 3.3: Toy example of set-based methods.

3.2.3 Video-based Methods

Video-based methods exploit the temporal coherence between adjacent frames in the video. For the VbOR problem, temporal coherence can be used to learn better representations from videos based on feature tracking [Noceti et al., 2009], or to remove

unstable local features [Sivic and Zisserman, 2009]. In addition, applying video-based techniques may facilitate learning the variation among object parts, detecting outlier features, or compressing the representation of video data. An example is shown in Figure 3.4, where the trajectories can be extracted via tracking and later used for matching. Features not on these trajectories are considered outlier features, and thus, discarded.

Extracting temporospatial coherence is important in many tasks, such as action recognition, video surveillance and object tracking. However, it is not trivial to do so in an egocentric setting, since the camera can move in an arbitrary manner and the time-ordering does not reflect any characteristics of the object’s identity. Greater computational power is also consumed due to the additional tracking process.

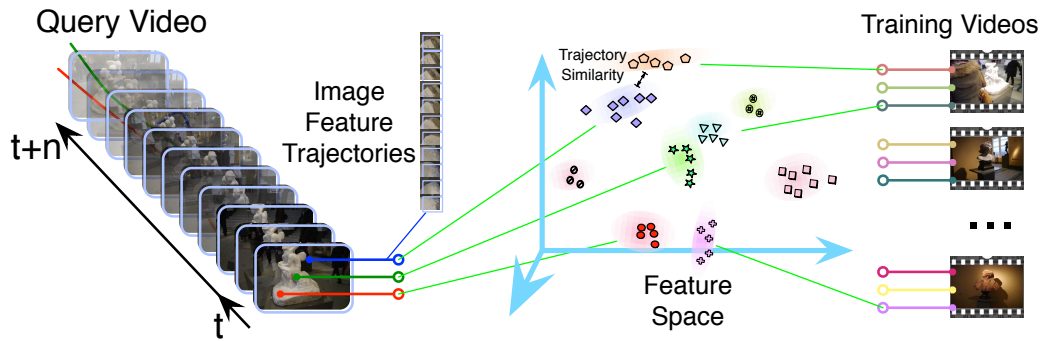


Figure 3.4: A toy example of trajectory matching methods based on feature tracking.

3.2.4 3D-based Methods

A different approach is to utilise 3D geometric cues. Earlier works such as [Funkhouser et al., 2003] required hand-crafted 3D CAD models as input, whilst in this study, we focus on more recent methods that reconstruct models from multi-view images, as shown in Figure 3.5. 3D-based object recognition is a relatively new yet attractive research field, which has been popularised by the emergence of low-cost depth cameras. In the case of rigid objects, 3D geometry can be treated as one of the most nuisance-

invariant cues that can be obtained from video.

In the literature, [Tangelder and Veltkamp, 2008] provide a comprehensive survey of how 3D CAD models can be used in content-based retrieval systems. Several recent studies, including object recognition [Hao et al., 2013b], landmark recognition [Irschara et al., 2009] and camera pose estimation [Sattler et al., 2012], have exploited 3D models reconstructed by photogrammetric methods, such as stereo matching [Hirschmuller, 2008] and structure-from-motion [Ummenhofer and Brox, 2012].

However, the use of 3D object models for object recognition has several limitations. Photogrammetric methods require camera calibration to retrieve the absolute scale and location of an object, and the 3D point cloud generated is generally sparse, and requires more computation. Moreover, the object is often required to be static in the scene.

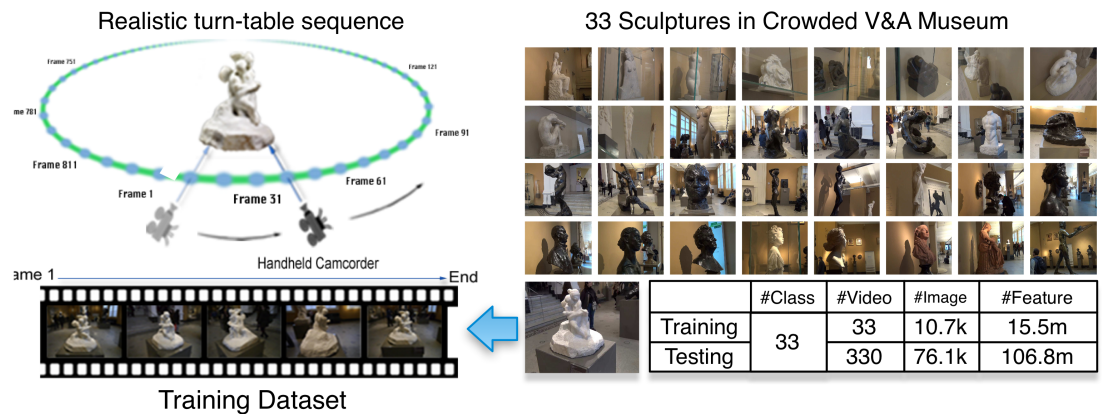


Figure 3.5: Toy example of 3D-based methods where each video is treated as an unordered image set.

3.3 Implementation

As illustrated in Figure 3.1, we implemented baseline methods and extended them by adapting geometric and temporal validation techniques. Although there exist frameworks for invariances of image features and receptive field responses under more gen-

eral classes of visual transformations [Tuytelaars and Mikolajczyk, 2008, Lindeberg, 2013], in this work we restrict ourselves to scale invariance as implemented in standard SIFT [Lowe, 2004a] in all experiments.

3.3.1 Overview

This section presents an overview of implemented methods in this chapter.

The detailed and coherent implementation approach are appended in chapter Appendix at the end of thesis.

3.3.1.1 Image-based Methods

We adopted the framework proposed in [Lowe, 2004a] as the baseline for the P2P approach, the standard bag-of-words approach [Fei-Fei and Perona, 2005] for I2I, and *NBNN* [Boiman et al., 2008] for P2C. In P2P and P2C, geometric validation was achieved by strictly applying *RANSAC* [Fischler and Bolles, 1981b] to correspondences based on a perspective transformation. In I2I, we employed Spatial Pyramid Matching (*SPM*) with a uniformed grid (as in [Lazebnik et al., 2006]), and spatial consistency using *Video Google* [Sivic and Zisserman, 2009]. Additionally, extensions for each image-based method are implemented. In I2I, we replaced the clustering method (k-means) by sparse coding with max pooling, as described in [Yang et al., 2009], to reduce the error from vector quantisation. Apart from that, better distance functions were employed from [Sivic and Zisserman, 2009]. In P2C, we implemented *Local-NBNN* [McCann and Lowe, 2012] as a state-of-the-art version of *NBNN*.

3.3.1.2 Set-based Methods

A kernel approach, *Kernel Principle Angles (KPA)* [Wolf and Shashua, 2003], is implemented to compute the principal angles in the feature space as the basis of the manifold-based method. We collected bag-of-words representations of video frames as the feature set based on the assumption that images in a sequence are highly correlated so that they lie on a low-dimensional manifold, which spans the variations in the object.

3.3.1.3 Video-based Methods

For video-based methods, we tracked interesting points bidirectionally [Kalal et al., 2010] with *Kanade-Lucas-Tomasi feature tracker (KLT)* [Lucas et al., 1981]. Three video-based methods were evaluated in the *Local-NBNN* framework: unstable feature removal (*Filtering*), averaged-trajectory matching (*TM*) and trajectory matching by KPA (*TM+KPA*). Additionally, we added a recent method [Liu et al., 2014] into the comparison. In *Filtering*, only unstable feature points are rejected; in *TM*, each trajectory is encoded into a single feature vector by averaging its feature points; and in *TM+KPA* and *Liu et al.'s method* [Liu et al., 2014], *KPA* is applied to obtain trajectory similarity measurements.

3.3.1.4 3D-based Methods

According to the general framework of 2D-to-3D image classification systems described in the literature [Hao et al., 2013b, Collet et al., 2011, Gordon and Lowe, 2006b], we first reconstructed 3D object point cloud models from the training videos using

the *VisualSfM* toolkit [Wu, 2011], which is a structure-from-motion based photogrammetric modelling program, where foreground segmentation is used to cleanse the noisy 3D point clouds. As shown in Figure 3.5, each 3D point corresponds to a set of image features from video sequences during appearance-based feature matching and 2D-to-3D geometric validation can then be applied to constrain the correspondences to a rigid transformation. We apply a fast non-iterative method called *ePnP* (efficient Perspective-n-Point) proposed by [Lepetit et al., 2009] to solve the Perspective-n-Point problem for 2D-to-3D transformation estimation; and apply 3D *RANSAC* for 3D-to-3D estimation, following the method proposed by [Forsyth and Ponce, 2002].

3.3.1.5 Hybrid Methods

Furthermore, we propose a hybrid method combining a video-based and a 3D-based method to incorporate benefits from both. To avoid unnecessary experiments, we chose to combine the best method of each category, i.e., *Local-NBNN+TM* and *2D-to-3D+ePnP*, by empirical results (see Section 3.5). In practice, each object 3D point cloud and video trajectories corresponds to a set of similar features, which can then be encoded by simply averaging the set of feature vectors into a single representation for better *Local-NBNN* performance. After matching video trajectories to 3D object points, *ePnP* validation is performed between the trajectories' coordinates of each frame and object 3D point coordinates. Similar to the 3D-based methods, only the correspondences that pass geometric validation are taken into account for the later voting procedure.

3.4 Dataset

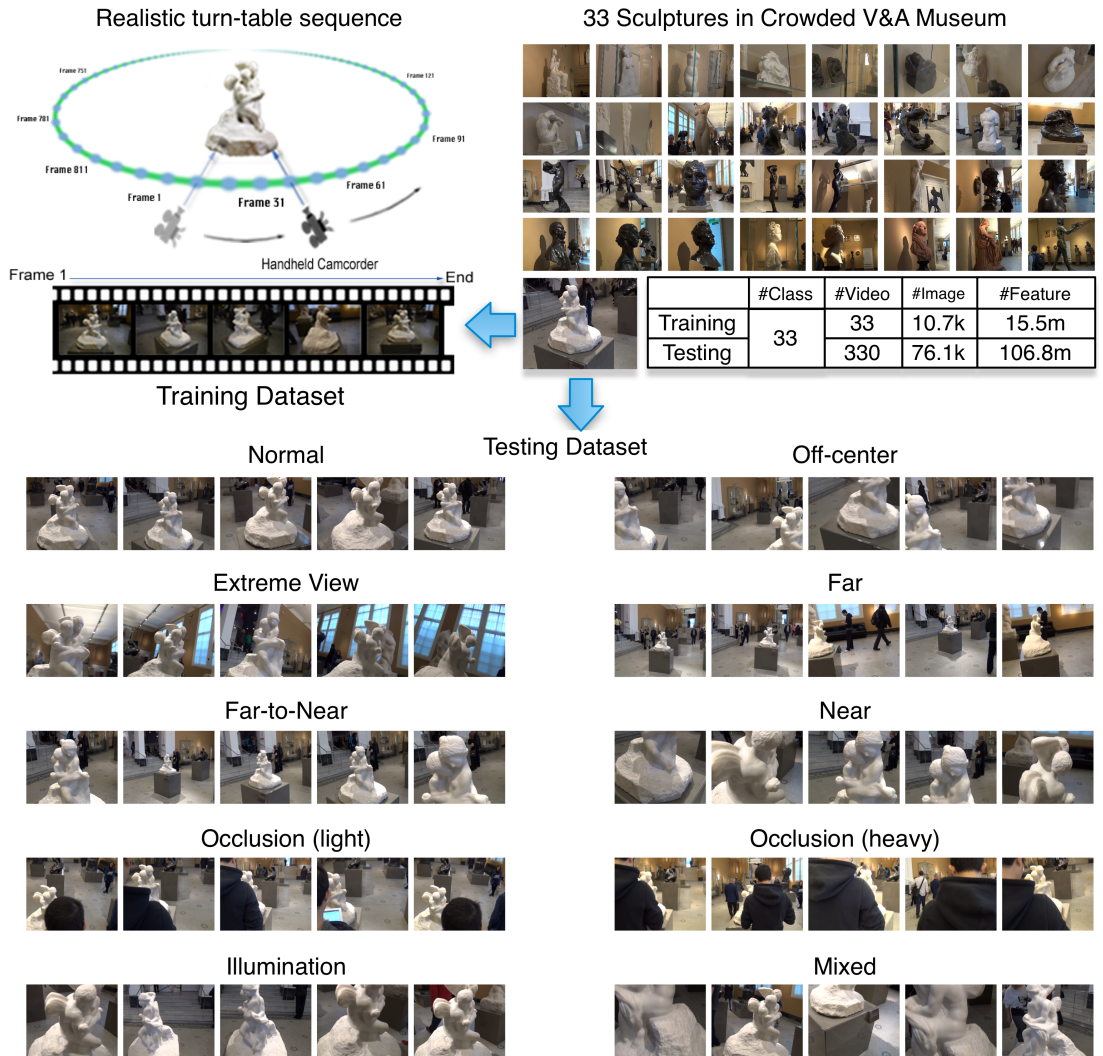


Figure 3.6: Illustration of the collected dataset.

Several egocentric datasets have been made publicly available [De la Torre et al., 2008, Ren and Philipose, 2009]. However, to the best of our knowledge, there is no suitable benchmark dataset for evaluating complex rigid object recognition methods on egocentric videos. Thus, we constructed a new video dataset with 33 less textured sculptures in cluttered museum scenes, as shown in Figure 3.6.

3.4. DATASET

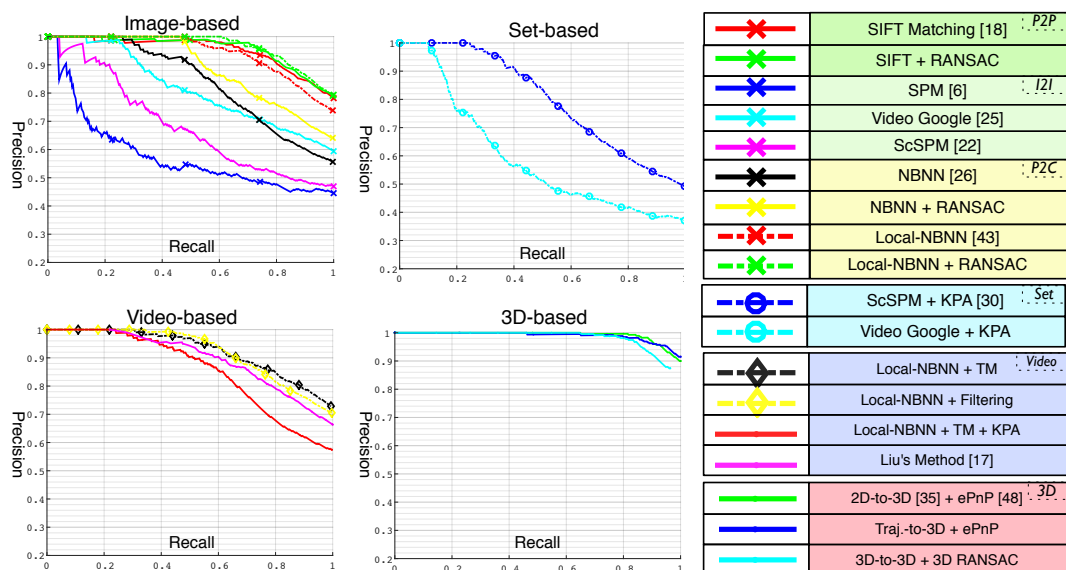


Figure 3.7: Full evaluation of the video classification results based on the precision-recall curves (all figures are best viewed in colour), including image-based methods (with voting), set-based methods, video-based methods and 3D-model based methods.

In total, 363 videos (30 fps, 720×576 pixels) were captured by amateur users with a hand-held camcorder in a crowded museum. 33 different sculptures served as object instances, each of which has a training video and 10 testing videos. The training videos were captured at 180 or 360 degrees from azimuth around the sculptures, depending on their positions. For testing videos, we deliberately added different nuisance including extreme views, large scale changes, occlusions, light reflection and temporal object disappearance.

Our proposed dataset is collected to solve real-life object instance recognition problem and has many unique properties compared to standard image or scene recognition datasets: (i) instance-level object classes; (ii) high inner-class correlation between frames within each video sequence; (iii) high inter-class correlation between object classes; (iv) videos are shot intentionally unprofessional with a set of predefined hand-

held camera movements and other additional nuisances. The significance of each property will be explained next.

While recent state-of-the-art computer vision systems are leveraging ‘big data’ to train a very complex model, an instance-level object recognition system is somewhat different. Most instance-level object recognition tasks are ‘category specific’, where object classes share some specific attributes, such as the dataset for automatic bin-picking usually consists of mechanical components with simple shape but no texture. In other words, a model trained with category-level dataset will be under-fitting and redundant for instance-level object recognition task. Furthermore, collecting and labelling a large instance-level dataset is not cost-effective since it is often too specific to be generalised to other tasks, since an object category may have infinite number of instance subsets. One exception is collecting huge dataset for face recognition system as all faces share many priorly known attributes and face recognition systems are highly demanded all over the world. That said, a model trained on a dataset that only contains western celebrities performs sub-optimally on asian pedestrians.

Next, the high inner-class correlation is naturally exist in video clips since the camera moves smoothly and captures the scene in a fast pace so adjacent frames are most likely similar. This temporospatial similarity can help us to track and collect more information from the scene despite the information is also highly redundant. Comparing with performing object recognition on a single camera shot, a well designed VbOR system should exploit temporospatial information while not wasting computation on the redundancies. Our proposed dataset poses realistic challenges to the VbOR systems. Each testing video clip consists of hundreds frames of an object under different types of nuisances, such as large camera movement, extreme view or occlusions that can cause failure in tracking.

It is obvious that the complexity of an image recognition task depends on the distinctiveness between each class in the set. A simple heuristic approach can easily

classify objects with different colour while recognising human faces requires a very complex convolutional neural network model with a modern advanced GPU. Our proposed dataset has only sculptures, all belong to one single object category.

There are several reasons that we believe the sculpture is a suitable object category for evaluating outlier removal methods and VbOR systems. First sculptures have several interesting attributes. They are large, rigid three-dimensional objects so strong geometric cues can be exploited for evaluating different approaches of outlier removal in VbOR systems. Also, sculptures are almost textureless and lack of colour variation. And this poses challenge on classic local feature descriptors that rely on the colour gradients. Moreover, some camera views can be even less informative such as the back of a portrait sculpture. This motivates the use of a video clip over a single image for object recognition, and evaluating temporospatial cues for accumulating information along the video.

Another motivation for proposing this dataset is that many public object recognition datasets for benchmarking or competition are automatically collected using on-line search engine, such as VOC object dataset [Everingham et al., 2010] and ImageNet [Deng et al., 2009], where a large amount of uploaded pictures on Internet are shot by professional cameramen. State-of-the-art object recognition methods train and test with the most advanced processors for a long time on millions of high quality images. In contrast, a real-life object recognition application is more likely to be deployed on a handheld device with very limited processing resource, and operated by amateur users.

Our proposed dataset aims to evaluate VbOR systems with various outlier removal methods to solve real-life problems. To serve this purpose, all videos in the dataset are collected by amateur camera users with additional realistic noises, such as camera motion blur and occlusions from museum visitors.

3.5 Evaluation

In Figures 3.7 and 3.8, an overview of all the experiments performed with our dataset is provided. The category of each experiment was determined by the representation of query testing videos. A table for coarse time consumption of all methods are illustrated in Figure 3.9, this is tested on a single core, 2.8 GHz Intel Core i7, 16 GB 1600 MHz DDR3.

Traditionally, object recognition evaluations report accuracy in percentage. To better demonstrate the impact of each method, we measured their performance in the form of a precision-recall (PR) curve, which was inspired by the ratio test in [Lowe, 2004a], since it can be easily generalised to evaluate the performance of image classification. For each query video, the assigned object class was deemed acceptable only if the ratio between the highest and second-highest class probability was above a certain threshold, otherwise, it was considered a false negative. If the highest class was the same as the ground truth, it was considered a true positive. By testing all possible thresholds, a full PR-curve is obtained. It is worth noting that some of the results obtained with image-based methods contrast their performance using public datasets. We consider this to be mainly due to the aforementioned uniqueness of our dataset.

Image-based Methods Only the final voting accuracy for each video is shown for image-based methods. In general, I2I methods had poor performance due to the quantisation of image descriptors as described in the literature [Boiman et al., 2008]. The larger (approximately 10k) visual vocabulary with a better distance function (Bhattacharyya distance) in *Video Google* or *SPM based on sparse coding (ScSPM)* improved the results obtained with *bag-of-words* to some extent, but it was still not as good as

3.5. EVALUATION

SIFT Matching	1.00	0.94	0.30	0.45	0.94	0.88	0.79	0.76	0.82	0.94
SIFT + RANSAC	1.00	0.94	0.36	0.48	0.94	0.88	0.82	0.79	0.85	0.88
SPM	0.82	0.64	0.33	0.21	0.61	0.30	0.45	0.24	0.52	0.33
Video Google	0.94	0.73	0.42	0.45	0.88	0.27	0.58	0.33	0.64	0.70
ScSPM	0.88	0.55	0.30	0.27	0.61	0.30	0.48	0.30	0.48	0.52
NBNN	0.94	0.73	0.18	0.39	0.64	0.52	0.52	0.48	0.61	0.55
NBNN + RANSAC	0.97	0.76	0.21	0.42	0.88	0.58	0.64	0.55	0.70	0.70
Local-NBNN	1.00	0.94	0.39	0.39	0.91	0.70	0.73	0.61	0.88	0.85
Local-NBNN + RANSAC	1.00	0.94	0.39	0.42	0.97	0.76	0.82	0.73	0.91	0.94
ScSPM + KPA	1.00	0.73	0.21	0.15	0.67	0.30	0.39	0.27	0.52	0.67
Video Google + KPA	0.82	0.33	0.33	0.18	0.52	0.27	0.39	0.15	0.39	0.30
Local-NBNN + TM	0.85	0.85	0.52	0.58	0.73	0.85	0.82	0.76	0.64	0.70
Local-NBNN + Filtering	0.82	0.85	0.52	0.58	0.70	0.79	0.76	0.70	0.64	0.70
Local-NBNN + TM + KPA	0.76	0.76	0.30	0.48	0.52	0.55	0.67	0.55	0.58	0.61
Liu's Method	0.97	0.76	0.33	0.48	0.85	0.58	0.73	0.58	0.58	0.79
2D-to-3D + ePnP	1.00	1.00	0.61	0.70	1.00	0.94	0.94	0.91	0.94	0.97
Traj.-to-3D + ePnP	1.00	0.97	0.67	0.73	1.00	0.94	0.97	0.97	0.94	0.97
	Normal	Off-center	Extreme View	Far	Far-to-near	Near	Occlusion (light)	Occlusion (heavy)	Illumination	Mixed

Figure 3.8: We divided our *V&A* dataset into 10 subsets by different types of nuisance. Each column represents one subset which contains 33 videos (one per class), whilst each row gives the results of a method. The numbers indicate the amount of successfully classified videos.

other methods.

The P2P and P2C methods achieved similar accuracy and outperformed I2I methods owing to the following reasons. Firstly, there was no feature quantisation in P2P and P2C and thus no loss of discriminatory power. Secondly, the geometric relationship among features is partially lost in I2I methods, whereas the robust estimation method *RANSAC* in P2P and P2C methods constrains the spatial distribution of image features, which is favourable for rigid object recognition.

Set-based Methods Overall, set-based manifold methods did not have significant

SIFT Matching	Real-time	SIFT Descriptor	FLANN	Voting		
SIFT + RANSAC	Below 10 FPS	SIFT Descriptor	FLANN	RANSAC	Voting	
SPM	Real-time	SIFT Descriptor	SPM Encoding	linear SVM		
Video Google	Real-time	SIFT Descriptor	FLANN	Spatial Consistency Voting		
ScSPM	Below 10 FPS	SIFT Descriptor	Sparse Coding	SPM Encoding	linear SVM	
NBNN	Real-time	SIFT Descriptor	FLANN	NBNN		
NBNN + RANSAC	Real-time	SIFT Descriptor	FLANN	NBNN	RANSAC	
Local-NBNN	Real-time	SIFT Descriptor	FLANN	L-NBNN		
Local-NBNN + RANSAC	Real-time	SIFT Descriptor	FLANN	L-NBNN	RANSAC	
ScSPM + KPA	Below 10 FPS	SPM (Set)	KPA			
Video Google + KPA	Below 10 FPS	Video Google (Set)	KPA			
Local-NBNN + TM	Real-time	SIFT Descriptor	Optical Flow	FLANN	L-NBNN	
Local-NBNN + Filtering	Real-time	SIFT Descriptor	Optical Flow	FLANN	L-NBNN	
Local-NBNN + TM + KPA	Below 1 FPS	SIFT Descriptor	Optical Flow	FLANN	L-NBNN	KPA (Trajectory)
Liu's Method	Real-time	SIFT Descriptor	Liu's Method			
2D-to-3D + ePnP	Real-time	SIFT Descriptor	FLANN (3D Point Cloud)	ePnP	Re-projection	
Traj.-to-3D + ePnP	Real-time	SIFT Descriptor	Optical Flow	FLANN (3D Point Cloud)	ePnP	Re-projection
3D-to-3D + 3D RANSAC	Below 1 FPS	SIFT Descriptor	Optical Flow	3D Reconstruction	3D RANSAC	

Figure 3.9: This table provides a coarse time consumption of each workflow of methods. The green colour indicates real-time, i.e. beyond 10 frames per second (FPS), or fast technique; The yellow colour indicates the method runs between 1 FPS to 10 FPS, or medium speed technique; The red colour indicates the method runs below 1 FPS, or slow technique.

impact on the performances of I2I methods. The main difficulty was related to the complex nuisance effects in scenes such as under extreme view or occlusions, since manifold-based methods are generally prone to set complexity and outliers. In addition, *KPA* [Wolf and Shashua, 2003] improves *ScSPM* [Yang et al., 2009], but degrades *Video Google* [Sivic and Zisserman, 2009]. The results showed that the advantage of applying *KPA* was reduced when the vocabulary size increased. The high heterogeneity of image representations caused the failure of *KPA* when determining dependencies within the set, thereby leading to inaccurate estimates of the subspace from the image set.

Video-based Methods The results of the comparisons between four methods, *Filtering*, *TM*, *TM+KPA*, *Liu et al.'s method* [Liu et al., 2014] and their baseline *Local-NBNN*, have shown that: (i) the formation of a trajectory did not improve recognition accuracy, (ii) averaging the trajectory obtained a similar performance, and (iii) *KPA* was computa-

tionally expensive and not suitable for application to trajectory matching.

These contradictory results can be explained as follows: (i) To reduce unstable features and prevent long-term drifting, a bidirectional validation was applied to the trajectories. As shown in Figure 3.11(c), approximately 90% of the features used in training and 67% in the testing dataset were filtered. However, the filtering process did not generate extra inliers and the inherent advantage of *Local-NBNN* is its robustness to noisy feature points, which explains their similar accuracy. (ii) Figure 3.11(a) shows that the trajectories were short due to the strict temporospatial constraint and unstable features increased due to the use of a hand-held camera. This feature is actually favourable to averaged trajectory matching, since the features in most of the trajectories are nearly identical, hence their mean is representative. (iii) However, this also explains the poor performance of the *KPA* approach because short trajectories were far from sufficient to span the variations in the object parts, thus lacking discriminatory power. In addition, it should be noted that the application of *KPA* is computationally intense with massive trajectories due to its high complexity.

However, by using trajectory averaging to detect outlier features, the dataset was compressed to 1.48% for training and to 5.30% for testing compared with their original sizes, as shown in Figure 3.11(c). This reduced the computational power and memory requirements, especially when using non-parametric classifiers.

3D-based Methods Retrieving the 3D mesh model of objects from the training video as in Figure 3.10, and performing 2D-to-3D geometric validation has increased the recognition performance dramatically for two reasons: (i) in training videos, the reconstruction process rejected features that were not consistent with the object geometry, such as pedestrians or specularities from light reflection, thereby resulting in a considerable increase in the signal-to-noise ratio of the database. (ii) Figure 3.11(b) shows the long-tail distribution of a number of features allocated to 3D points, which

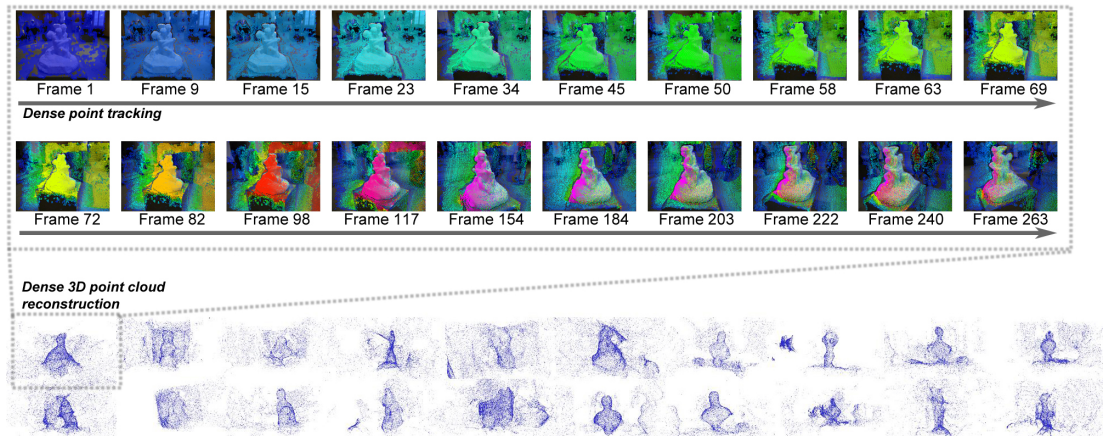


Figure 3.10: 3D object point clouds are reconstructed via structure-from-motion algorithms for 2D-to-3D geometric validation.

indicates that a considerable amount of 3D points contained features that covered a large range of view. The 3D geometry is viewpoint-invariant according to the assumption of the objects rigidity, thus the 2D-to-3D geometric validation strictly constrained the correspondences, which resulted in a higher confidence in the final voting of the object class. This is especially helpful for VbOR because the object class can usually be determined from a few confident frames within the video. Furthermore, the training dataset was compressed to 67.23% after 3D reconstruction and to 4.75% after it was averaged further into a single feature vector, according to Figure 3.11(c). 3D-to-3D also achieved good recognition accuracy, but the reconstruction process during the testing stage required too much memory and computational power, which made it inefficient compared to 2D-to-3D methods.

Proposed Hybrid Methods The combined method exploited the advantages of 3D-based and video-based methods, and achieved one of the best P-R rates, as shown in Figure 3.12. It also compressed the training dataset to 4.75% by averaging the features in each 3D point and compressed the testing dataset to 5.30% by averaging the trajectories, leading to significant reduction in memory consumption whilst maintaining the

3.5. EVALUATION

discriminatory property within the target objects.

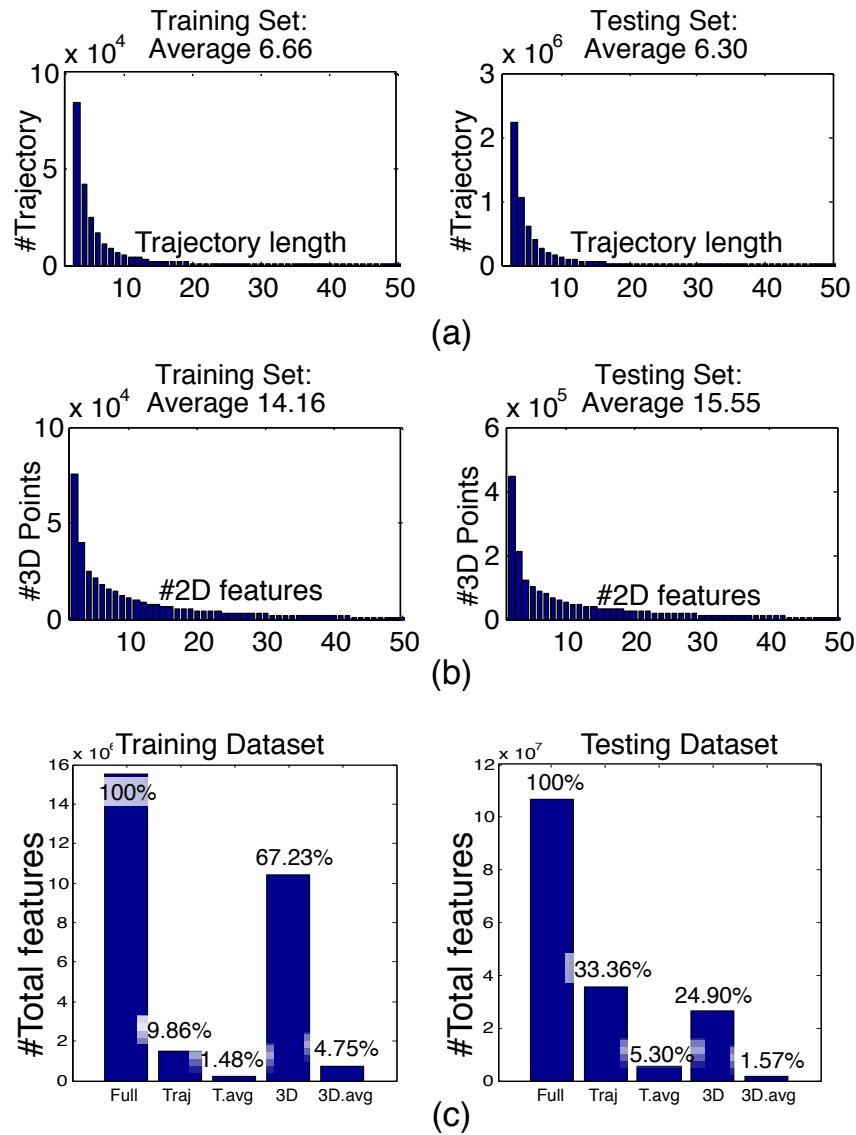


Figure 3.11: Histogram showing (a) the number of frames crossed per trajectory and (b) the number of SIFT features allocated per 3D point. (c) The remaining percentage of features after outlier removal via different methods.

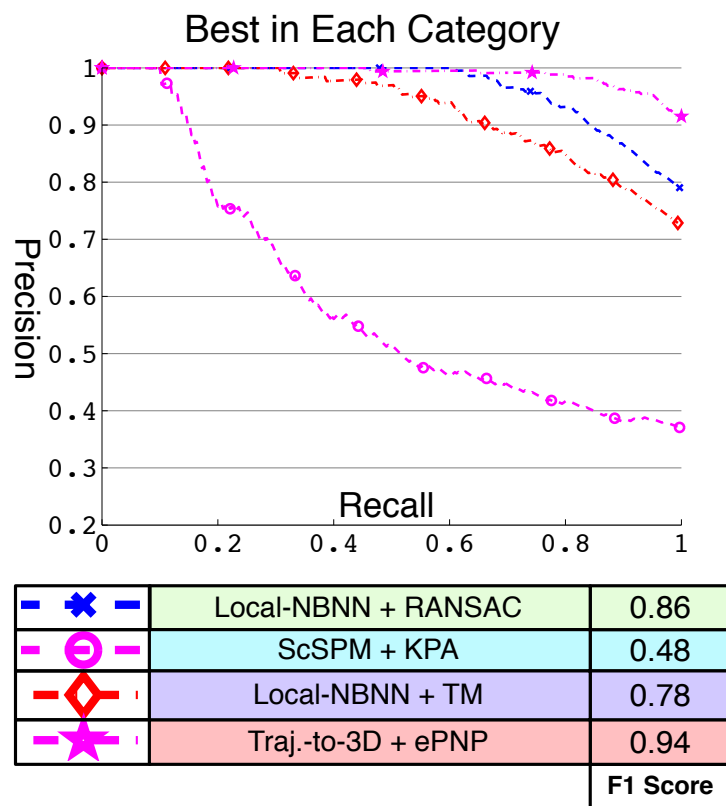


Figure 3.12: Precision-recall curves with the hybrid method compared with the best method from each method category.

3.6 Summary

This chapter aims to study the use of outlier removal methods to enhance video-based object recognition systems.

Firstly we proposed a new video dataset that is collected in a crowded museum for evaluating object instance recognition systems under realistic environment. The dataset poses difficulties on the classic image-based object recognition systems and hence motivates the use of their video-based extension, which outlier removal methods play important roles.

We then categorised and compared diverse state-of-the-art object recognition frameworks and their video-based extensions. To have fair comparison and focus on the extending outlier removal approaches, all frameworks based on the classic SIFT interest point detector / feature descriptors despite there are much more up-to-date techniques.

Based on the empirical evaluation results, we conclude that with video- and 3D-based extensions, several frameworks outperformed their image-based baseline in both recognition accuracy and efficiency. These extensions are largely benefit from the outlier removal methods based on two important cue from multi-view video sequences: a temporospatial cue that allows us to apply tracking technique to remove unstable interest point proposals and help to build another important geometric cue, which poses a strict 3D geometric constraint to further remove false proposals and validate the 3D object shape.

We have found that the most promising method for achieving the best object recognition performance on egocentric videos is to clean the training set with geometric constraints and classify the query videos with geometric cue and temporospatial cue. Thus we proposed a novel hybrid method by combining both cues to achieve the best

performance.

3.6.1 Generalisation

From this work, the result shows that for rigid object recognition task, first we can easily build the object model from a single turn-table video sequence via 3D reconstruction techniques, such as Structure-from-Motion, or SfM. Then we can leverage 3D object models with 2D-to-3D geometry validation techniques to greatly boost the recognition accuracy. This workflow requires little human effort. It is also simple to implement, easy to swap any technique involved to the state-of-the-art techniques and can be generalised to most of rigid object recognition systems, such as for bin-picking robot or cashier-free grocery store.

From the aspect of machine learning, if outlier removal methods are applied properly, they can efficiently extract information-of-interest from a very redundant source. The idea presented in this chapter can be generalised to other machine learning systems as long as the ‘uniquely shared’ attributes can be found, either heuristically or by machine learning techniques. In the case of video-based object recognition, temporo-spatial and geometric cues are the distinctive attributes shared in all positive feature proposals.

However, even the machine learning techniques nowadays are robust to the data distribution, the cost to auto-learn the latent attributes could be extremely high, sometimes may require much more training data or need intensive data permutation. One example would be the deep convolutional neural networks (CNN). There are million of parameters required to cover all varieties of data. Also it usually requires hundreds of thousand data pieces with extra data permutation to train. From this study, we find that Space Pyramid Matching (SPM) performs poorly due to the spatial inconsistency between training and testing videos, despite it achieves very good result on scene

recognition dataset.

Thus, we defend the importance of heuristic or human knowledge to guide machine learning tasks, and to exploit them as an outlier removal preprocessing module to boost the overall system performance, despite it is generally not favourable in the academia. Even for the tasks involving CNN, such as commercial-level face recognition system ([Deng et al., 2018]), it still heavily involves human designed workflow to achieve satisfactory result, such as using face detector and face alignment as preprocessor to standardised faces. Another example is human attribute recognition ([Bourdev et al., 2011]): with a human pose estimator as preprocessor to locate the body keypoints as an extra input, the recognition performance will largely raise. However, this guidance might be replaced by meta-learning techniques in the future.

3.6.2 Next Chapter

Next chapter presents a learning-based approach to perform outlier removal to assist 3D object pose estimation.

4

CHAPTER

REAL-TIME BACKGROUND-AWARE 3D TEXTURELESS OBJECT POSE ESTIMATION

CONTENTS

4.1 Overview	63
4.2 Related work on 6-DoF pose estimation	65
4.3 Method	69

4.1 Overview

In this chapter we propose a learning-based outlier removal approach for real-time 3D object pose estimation. We present a modified fuzzy decision ternary forest that trains on typical template representation. We employ an extra preemptive background rejector node in the decision forest framework to terminate the examination of background

locations as early as possible, result in a significantly improvement on efficiency. This approach is also scalable to a large dataset since the tree structure naturally provides a logarithm time complexity to the number of objects. Finally, we further reduce the validation stage with a fast breadth-first scheme. The results show that our approach outperforms the state-of-the-art approaches on efficiency while maintaining a comparable accuracy.

In this work, we focus on 3D rigid textureless object pose estimation with RGB-D data. Since rigid object pose has 6 degrees of freedom ($x, y, z, \text{pitch}, \text{yaw}, \text{roll}$), this problem is also known as 6-DoF pose estimation. One typical method of 6-DoF pose estimation is through *template matching*, which requires a mesh model that is usually obtained via scanning the target object. A large set of annotated training ‘templates’ can then be generated by rendering to uniformly cover the pose space. During test time, a sufficiently similar template is found via a distance-based search process, often via approximate nearest neighbour (ANN) techniques.

There are mainly three different ways of performing nearest neighbour searching on template matching: exhaustive, hashing-based and tree-based. Although given the feature descriptor, exhaustive methods (e.g. , LineMOD [Hinterstoisser et al., 2012a]) guarantee finding the most similar match, its linear complexity is definitely not ideal. Hashing-based methods, on the other hand, have sublinear or even constant complexity during searching. However, the design of an efficient hash function with good trade-off between memory consumption and matching performance is not trivial. Using tree-based methods to solve ANN problem, e.g. , k -d tree, can also significantly lower the complexity. However, the cascade nature of tree structure suffers from strong outlier rate due to error accumulation. The efficiency suffers as well from the curse of dimensionality due to *backtrack*.

Our approach aims at a further speedup in state-of-the-art template-matching methods for real-time applications, while maintaining a comparable pose estimation

accuracy. Targeting the bottlenecks of a typical template matching process, we extend the classic randomised decision forest framework in several perspectives for acceleration. Specifically, we focus on alleviating the computation spent on background noises and filtering false positives. Instead of simply applying a data-driven objectness detector as a standalone preprocessing stage, we achieve a better outcome by attaching an additional node as ‘background rejector’ to each split node in the decision tree. An additional minor contribution is an adaptation of a RANSAC-based algorithm [Nistér, 2005a] into final validation stage to allow a further speed up.

The merit of this approach is to reject false candidates as early as possible and also pass a good candidate to the validation stage in one go. This approach has the benefits of both the holistic template and the tree structure. As a holistic template, it focuses more on global information than local descriptors, and tends to capture object shape rather than texture. Since a specific object pose is defined by a single distinctive template, it also does not necessarily require a computationally expensive geometric validation step. Secondly, by constructing a decision forest, similar views are clustered hierarchically for a faster search time, but also allows us to exploit them for background rejection. The experiment result shows superior speed and sublinear complexity with comparable accuracy to the state-of-the-art approaches.

4.2 Related work on 6-DoF pose estimation

Existing 6-DoF pose estimation methods can be categorised into distance-based, learning-based and registration-based methods.

Distance-based methods approach this problem by defining a distance metric to measure the similarity between samples. Then a set of samples from different view points, usually rendered from 3D object models, is generated as the training set. Dur-

ing testing, the object pose on a query image is retrieved by pairwise comparison between extracted templates and the training set.

To effectively measure the similarity between object views, a compact and discriminative description vector is required. [Hinterstoisser et al., 2012b] present a novel image representation, a rigid template using colour gradient and surface normal as feature descriptors called LineMOD. The templates are synthetically rendered from 3D object mesh models under different scales and view angles. Similar to other traditional template matching approaches, each template matches with all possible locations across the query image to produce a similarity score map. Despite the exhaustive search, it achieves real-time speed for single object pose estimation.

Tree-based approaches apply binary search in multidimensional space. Given a query point and a set of data points, this approach partitions the search space roughly into halves in each iteration, until there is only one data point left in the search space. The complexity is therefore $O(\log n)$ to the number of data points. K-d trees are generally considered unsuitable for high-dimensional spaces searching as most of the points in the tree will be evaluated and the efficiency is no better than exhaustive search [Goodman et al., 2000]. One improvement by [Beis and Lowe, 1997], called best-bin first algorithm, uses a backtracking strategy to prioritise the searching queue based on closeness and achieves two orders of magnitude speed up. Another solution applies randomness in building multiple trees to improve the search speed at the cost of the individual k-d tree not always returning the exact nearest neighbours.

Hash table is a well-known data structure that allows a symbol lookup in $O(1)$ complexity. In other words, the searching time is constant regardless of the database size. However, a hash table can only be able to find the exact match while in ANN searching problem, we seek approximate matches. The most straight forward solution is hashing the whole quantised feature space into a single hash table so that every possible query point directly maps to their nearest data points. Unfortunately this naive approach

is no longer feasible for high-dimensional data. A recent work by [Kehl et al., 2015] employed hashing techniques to achieve sublinear scalability by exploring different hashing key learning strategies and achieving sublinear complexity to the number of templates, outperforming the state-of-the-art methods in terms of runtime.

Learning-based methods usually generalise better to variations in viewpoint, translation and slight shape deformations.

The methods that fall into this category focus on better generalisation to slight variations in translation, local shape and viewpoint. The explicit background/foreground separation is learnt parametrically to deal with heavy background clutter. The result shows that these approaches cause less false positives than nearest neighbour approaches. However, the efficacy is their dependency on the quality of negative training samples, and this benefit may not transfer across different domains. Tejani et al. [Tejani et al., 2014] propose incorporating a one-class learning scheme into the hough forest framework for 6-DoF problems. Rios-Cabrera and Tuytelaars [Rios-Cabrera and Tuytelaars, 2013] extend LineMOD by learning the templates in a discriminative fashion and handle 10-30 3D objects at frame rates above 10fps using a single CPU core.

Registration-based methods attempt to fit a pose hypothesis to the observation, by iteratively updating and minimising the discrepancy between the query sample and a sample rendered from the current pose hypothesis. A popular choice is the Iterative Closest Point (ICP) [Fitzgibbon, 2003].

Johnson and Hebert present an early seminal work [Johnson and Hebert, 1999] for simultaneous recognition of multiple objects in scenes containing clutter and occlusion, based on matching surfaces by matching points using the spin image representation. [Gordon and Lowe, 2006a] present a feature-based object pose estimation framework that accurately tracks the camera using learned models and SIFT features [Lowe, 2004b]. The estimation is performed by matching query image features with 3D object

model features and solving the Perspective-n-Point (PnP) problem for the 2D-to-3D correspondences. Drost et al. [Drost et al., 2010] propose a novel method that creates a global model description based on oriented point pair features and matches that model locally using a fast voting scheme. Another recent work [Hao et al., 2013a] improves the framework by introducing a novel matching scheme.

4.2.1 Related Work on Accelerating Template Matching

The first difficulty of accelerating template matching with efficient searching schemes comes from the high-dimensionality. One single coordinate is not representative enough to quickly reject candidates, thus leading to suboptimal performance. There are recent works have shown that with a well-chosen feature descriptor, few coordinates are reliable enough to find the match. LineMOD [Hinterstoisser et al., 2012a] achieves good performance by extracting only the best one hundred dimensions out of ten thousand from each individual template to perform an optimised exhaustive search. However, it is not trivial to apply an efficient searching algorithm based on this approach. Since the ‘best’ feature dimensions are in different subspaces for different objects, so the distance measurement between them is not meaningful, and ten thousand dimensions are simply too large for most of ANN algorithms.

One feasible solution is to cluster the templates into few sets, which has been proposed in a few recent works. Hashmod [Kehl et al., 2015] clusters the templates with a randomised decision forest and employs hashing techniques; Discriminatively Trained Templates (DTT) [Rios-Cabrera and Tuytelaars, 2013] cluster the templates with a bottom-up clustering method and construct strong classifiers using AdaBoost. The underlying reason is that the clustered subsets share common ‘relevant’ feature dimensions, that is to say, the templates in a subset can be well-classified using fewer coordinates.

Another difficulty is the heavy noises present due to the background clutter, occlusion and other environmental nuisances. Since the features in a template are generally local to a small region, it is very likely that the noises render some feature dimensions completely irrelevant to the ground truth. It is necessary to examine multiple feature dimensions to increase the signal-to-noise-ratio, thus producing reliable matching results.

With the same reason, a final validation stage is inevitable for achieving a good precision-recall rate. At this stage, a full similarity measure is calculated between the testing image region and a small subset of templates. Compared to previous stages, validation is expensive and is usually the bottleneck of the whole method, due to many more feature dimensions being involved in the calculation. Therefore, to reduce the computational cost, a good trade-off needs to be made between the size of validation subset and matching accuracy.

To sum up, a good approach to accelerate template matching should be able to: (i) ignore irrelevant feature dimensions; (ii) test on multiple dimensions simultaneously to be less prone to outliers; (iii) reduce the validation subset as much as possible while maintaining the matching robustness; and (iv) optimise the validation process to further speed up. With these in mind, we propose our tree-based method to address the problems for efficient template matching.

4.3 Method

Decision forest is one of the most commonly known framework for sublinear nearest neighbour search. However, training a tree-based classifier using templates directly is problematic due to insufficient training samples and noisy feature dimensions. In recent works on 3D object pose estimation using tree-based classifier, both [Tejani et al.,

2014, Brachmann et al., 2014] use local feature-based representations instead of holistic templates to alleviate the overfitting issue. However, this approach requires an additional geometric verification stage, Hough voting in [Tejani et al., 2014] and RANSAC-based optimisation in [Brachmann et al., 2014], that are likely to drag the process out too long to meet the requirement of real-time applications.

In this work, we propose several extensions of the decision forest framework to significantly accelerate template matching with marginal loss in accuracy compared to exhaustive search.

4.3.1 Template Matching with Decision Forest

The template matching approach describes each view of the object instance into a single template representation, defined as $\mathcal{T} = (O, \mathcal{U})$, where O is a reference RGB-D image of an object and \mathcal{U} denotes the set of locations u in O . The similarity measurement \mathcal{E} between a template \mathcal{T} and an input image I shifted by c can be formalised as:

$$\mathcal{E}(I, \mathcal{T}, c) = \sum_{u \in \mathcal{U}} \|(\Psi(O, u) - \Psi(I, c + u))\|, \quad (4.1)$$

where Ψ denotes the local feature descriptor, $\|\cdot\|$ denotes the distance function. Thus, the overall similarity is the sum of all individual corresponding local features differences.

Given an input RGB-D image, we use randomised decision forest $\{\mathcal{P}\}$ to classify sliding windows centred at each pixel location c . The leaf node of each tree \mathcal{P} that the pixel ends up in retrieves a set of template $\{\mathcal{T}\}$ so that a final classification is produced by a full validation.

Training Similar to most recent approaches on 6-dof textureless object pose estimation problem, we synthetically generate our template dataset from 3D object CAD models.

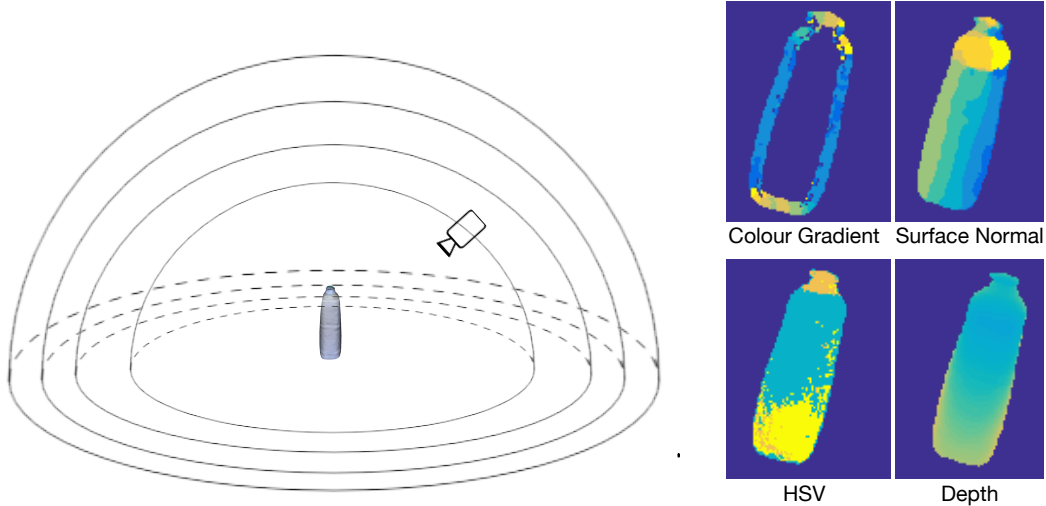


Figure 4.1: We generate synthetic dataset from object models that provided by LineMOD dataset. The left figure shows a sample procedure of rendering templates on a hemisphere of several radii. Here we use four modalities in this work, from left-top: colour gradient, surface normal, hue colour and depth.

The templates are computed from rendered object views on upper hemispheres of several radii, same as the sampling strategy in [Hinterstoisser et al., 2012b], as shown in Figure 4.1. Each template \mathcal{T} is assigned with a tuple label $(obj, rotation)$, which denotes the object class and object rotation (yaw, roll and pitch angles) respectively. Note that since the method is sliding-window based, object location can be retrieved from the position of the window, and thus we have 6-DoF pose of the object instance.

We first expand the template \mathcal{T} into a $|\mathcal{U}|$ -dimensional descriptor: $v_{\mathcal{T}} = \{\Psi(O, u) : r \in \mathcal{U}\}$. In practice, we use LineMOD as our descriptor including an additional object hue map as described in [Hinterstoisser et al., 2012b]:

$$\Psi(O, u) = \{CG(O, u), SN(O, u), HUE(O, u)\}, \quad (4.2)$$

where CG, SN and HUE represent three modalities used in LineMOD: colour gradient, surface normal and hue channel respectively. Each template is therefore a high dimensional vector of integers: $v = (x_{(1,1)}, \dots, x_{(|\mathcal{U}|, |M|)})$, where M is a list of modality

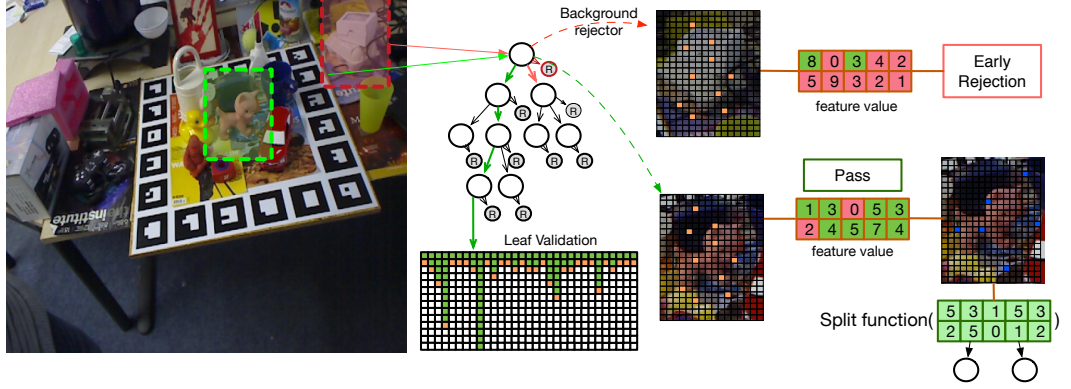


Figure 4.2: Visualisation of our proposed pipeline. At each candidate sliding window, we extract LineMOD feature descriptor and pass to the decision tree. At each node, the extracted features are examined by a preemptive background rejector, the candidate enters full validation stage only if it passes all the rejectors. This process significantly save the time cost on background noises. Finally, we further accelerate the validation stage with a fast breadth-first scheme, inspired by [Nistér, 2005a].

used and x is quantised feature value from 0 to 8. The value 0 appears when the feature is not significant, i.e. the magnitude of extracted feature is below a certain threshold. See [Hinterstoisser et al., 2012b] for details of each modality. For the feature outside object mask, we use uniform noise to model the background. Different background models are evaluated in [Brachmann et al., 2014].

Split Function A template descriptor set \mathcal{S}_n , arriving at n th node, is partitioned into two subsets $\mathcal{S}_n^L, \mathcal{S}_n^R$ by a split function $h(v, \theta) \in \{0, 1\}$:

$$\begin{cases} \mathcal{S}_n^L(\mathcal{S}_n, \theta) &= \{v \in \mathcal{S}_n | h(v, \theta) = 0\} \\ \mathcal{S}_n^R(\mathcal{S}_n, \theta) &= \{v \in \mathcal{S}_n | h(v, \theta) = 1\} \end{cases} \quad (4.3)$$

where θ denotes split node parameters. The split node parameter can be denoted as $\theta = (\phi, \psi, \tau)$, where ϕ selects a small subspace of entire feature space as feature selector function, ψ defines the geometric primitive used to separate the data, τ denotes thresholds in the binary test. The parameter is chosen to maximise an energy function, usually the information gain, to ensure an optimal split. In practice, the design off the

split function is crucial to achieve good performance. In a later section, we will discuss the impact of different split functions on template matching performance.

Leaf Validation The training templates are recursively split until it meets stopping criteria. This involves the control of tree shape, depth and thus, the trade-off between the generalisation power and efficiency. As briefly explained in the previous section, full validation on a template set is expensive and almost always the bottleneck of the whole pipeline, as in many recent related works. Ideally, we want to keep the number of templates in leaf nodes as few as possible while avoiding overfitting.

In practice, when we apply tree-based search on standard template matching, no matter how we set the stopping criteria, the lack of training data and high dimensional features is likely to lead to overfitting. One possible workaround is a variant of k -d tree approach (Best-bin-first) [Beis and Lowe, 1997], which backtracks from the leaf node according to a priority queue based on the closeness between query and the bin boundary, until a fixed number of nearest candidates is searched. However, this method is less efficient when large outliers are present, as the closeness is no longer reliable. Also, the optimal number of nearest candidates from backtracking varies with object class and can only be decided empirically. We will also address this issue later in our method.

4.3.2 Split Function for Insufficient and Noisy Data

In many applications, feature selector $\phi(v) \in \mathbb{R}^{d'}$, where often $d' = 1$ or 2 is sufficient. However, more dimensions are needed to compensate for the less distinctive, heavily quantised features and higher outlier ratio.

Figure 4.2 shows an overview of our method. To avoid the superlinear time cost from randomised node optimisation due to high dimensionality, we randomly draw

an exemplar v_e from the template set \mathcal{S} , such that:

$$h(v, \theta) = \begin{cases} 1, & \text{if } (\|\phi(v_e) - \phi(v)\| < \tau) \\ 0, & \text{otherwise} \end{cases} \quad (4.4)$$

$$\phi(v) = (x_{\phi_1}, x_{\phi_2}, \dots, x_{\phi_d}), \phi_i \in [1, d]$$

which maximises the energy function E :

$$\theta_n = \underset{(\theta \in \Theta)}{\operatorname{argmax}} E(\mathcal{S}_n, \theta), \quad (4.5)$$

where E denotes the energy function, Θ denotes a randomly generated set from the entire parameter space. Since the space is greatly reduced with the exemplar approach, the size of Θ should be limited to a small number to maintain the efficiency.

For the choice of energy function, we modify the standard entropy function to cope with the missing feature values:

$$E(\mathcal{S}, \theta) = \sum_{v \in \mathcal{S}} \lambda(\phi(v)) * \mathcal{I}(\mathcal{S}, \theta) \quad (4.6)$$

$$\lambda(v) = \sum_{x \in v} \gamma(x) \in \{0, 1\}$$

$$\mathcal{I}(\mathcal{S}, \theta) = \mathcal{H}(\mathcal{S}, \theta) - \frac{|\mathcal{S}^L| \mathcal{H}(\mathcal{S}^L, \theta) + |\mathcal{S}^R| \mathcal{H}(\mathcal{S}^R, \theta)}{|\mathcal{S}|}$$

where γ denotes a foreground Boolean indicator such that $\gamma(x) = 1$ if x located on the object mask on the template and vice versa; \mathcal{H} denotes an entropy function. In template matching or NN problem in general, each data point is assigned to a unique label. Therefore the standard entropy function is not suitable here. Instead we use an

unsupervised variant:

$$\begin{aligned} \mathcal{H}(\mathcal{S}, \theta) &= - \sum_{i \in \mathcal{V}} D(\mathcal{S}, i, \theta) \log D(\mathcal{S}, i, \theta) & (4.7) \\ D(\mathcal{S}, i, \theta) &= \frac{1}{|\mathcal{S}| |\phi(v)|} \sum_{v \in \mathcal{S}} \sum_{x \in \phi(v)} \delta(x, i) \\ \delta(x, i) &= \begin{cases} 1, & \text{if } (x = i) \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where \mathcal{V} is the feature space, i.e. $\mathcal{V} = \{1, 2, \dots, 8\}$ in the case of LINE-MOD descriptor. The entropy measures the uncertainty associated with the feature values given the feature dimension, a higher entropy yields better separation.

Next, we adapt a simple fuzzy rule to the thresholding to deal with insufficient data. This approach has been proposed in the literature [Olaru and Wehenkel, 2003, Yuan and Shaw, 1995], but has not drawn much attention in the field of computer vision. This adaptation tends to tolerate imprecise, missing feature values and reduce classification ambiguity from the split function, achieved by duplicating testing sample to both child nodes if it is too close to the split subspace.

We modify the binary test in Equation 4.3 and 4.4 such that:

$$\begin{cases} \mathcal{S}_n^L(\mathcal{S}_n, \theta) &= \{v \in \mathcal{S}_n | h(v, \theta) < \xi\} \\ \mathcal{S}_n^R(\mathcal{S}_n, \theta) &= \{v \in \mathcal{S}_n | h(v, \theta) > -\xi\} \end{cases} \quad (4.8)$$

$$h(v, \theta) = \|\phi(v_e) - \phi(v)\| - \tau, \quad (4.9)$$

Thus, feature vectors that fall into the ‘fuzzy’ interval $[-\xi, \xi]$ will be passed to both child nodes. This approach allows feature vectors to reach multiple leaves, which greatly reduces the overfitting due to lack of training data.

4.3.3 Preemptive Background Rejector

A fast coarse estimation of objectness is common in many detection methods, since the object of interest generally occupies only a small portion of the testing image. We further propose a preemptive background rejector as an extra split function in each node that sends the query to a ‘background’ leaf node if it fails a binary test. In contrast to most of the background removal methods, our approach does not exploit negative samples. Instead, we make an assumption that all feature vectors that do not exist in the dataset are negative samples. Here, we isolate the foreground from the background by minimising the entropy in the rejector function, so that all foreground feature vectors share similar values:

$$\check{h}(v, \check{\theta}) = \begin{cases} 1, & \text{if } (\frac{1}{|\phi(v)|} \sum_{x \in \phi(v)} \beta(\mathcal{S}_n, x, \check{\theta}) < \check{\tau}) \\ 0, & \text{otherwise} \end{cases} \quad (4.10)$$

$$\check{\theta}_n = \operatorname{argmin}_{(\check{\theta}_n \in \Theta)} \sum_{v \in \mathcal{S}} \mathcal{I}(\mathcal{S}, \theta), \quad (4.11)$$

where $\check{\tau}$ denotes a threshold in $[0, 1]$ to control the acceptance of outlier ratio; β denotes a background feature look-up table, such that:

$$\beta(\mathcal{S}_n, i, \check{\theta}) = \begin{cases} 1, & \text{if } D(\mathcal{S}_n, i, \check{\theta}) > \rho \\ 0, & \text{otherwise} \end{cases} \quad (4.12)$$

The query v is rejected immediately at a node n if $\check{h}(v, \check{\theta}_n) = 1$.

4.3.4 Fast Breadth-First Leaf Validation

The leaf nodes contain only tens, or at most, a hundred templates; however, pairwise matching all candidates is still computationally expensive. In practice, most bad candidates can be safely removed by examining only a small portion from the whole feature

descriptor. Therefore we propose a further speedup of the validation process with a breadth-first preemption scheme inspired by preemptive RANSAC [Nistér, 2005a].

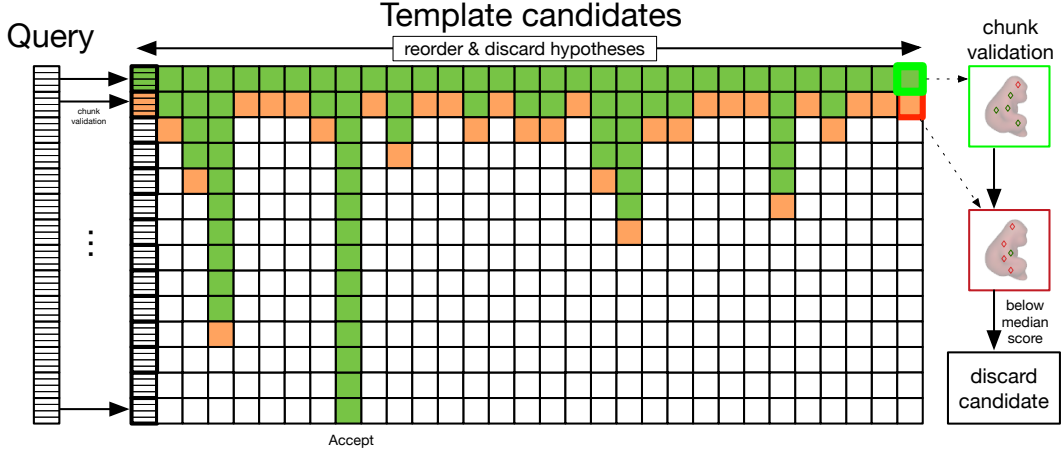


Figure 4.3: Breath-first preemptive scheme for leaf validation speed up. The templates are equally split into small chunks to alleviate the time cost from validating bad candidates.

As shown in Figure 4.3, during leaf validation, we equally split template feature descriptors into smaller chunks, and each contains a fixed number of features. In each stage, we score and compare all chunks and keep only the candidates that satisfy the pass condition. Scores are accumulated to the next stage and repeat until there is only one or no candidate left.

In our real-time implementation, we use the pass condition

$$f(v') = \begin{cases} 1, & \text{if } (s(v') > \max(\text{Median}(s(v)), \alpha)) \\ 0, & \text{otherwise,} \end{cases} \quad (4.13)$$

where $s(v)$ is a scoring function that measures the distance between query and candidate, α is a constant threshold. We set the threshold $\alpha = 0.5$ empirically. With this approach, the validation time complexity is reduced to $\mathcal{O}(\log n)$. The chunk size works as a trade-off between accuracy and speed: larger chunk leads to better robustness but

less efficiency, and vice-versa. Furthermore, we add depth value as another modality in the validation stage to measure shape similarity.

Table 4.1: Accuracy and average time per frame for the whole pipeline with 1, 5, 20 and 50 trees. Our approach is several times faster than the state-of-the-art approaches with comparable accuracy. T_Tree, T_Valid and T_Total are the time cost per frame from decision tree, leaf validation and overall respectively.

	1 Tree				5 Trees			
	T_Tree	T_Valid	T_Total	Acc.	T_Tree	T_Valid	T_Total	Acc.
ape	0.20 ms	6.50 ms	6.70 ms	96.0%	0.99 ms	12.31 ms	13.30 ms	97.1%
bvise	0.43 ms	13.37 ms	13.80 ms	91.1%	2.13 ms	29.50 ms	31.63 ms	93.2%
cam	0.41 ms	11.70 ms	12.11 ms	93.1%	1.93 ms	24.66 ms	26.59 ms	94.8%
can	0.44 ms	13.07 ms	13.51 ms	91.5%	2.22 ms	28.45 ms	30.67 ms	92.0%
cat	0.23 ms	8.34 ms	8.57 ms	94.3%	1.05 ms	15.23 ms	16.28 ms	95.5%
driller	0.39 ms	13.93 ms	14.32 ms	95.4%	1.73 ms	29.01 ms	30.74 ms	96.0%
duck	0.28 ms	9.64 ms	9.92 ms	90.0%	1.20 ms	15.99 ms	17.19 ms	94.5%
eggbox	0.30 ms	10.73 ms	11.03 ms	98.3%	1.18 ms	20.13 ms	21.31 ms	98.9%
glue	0.34 ms	11.46 ms	11.80 ms	92.1%	1.62 ms	29.85 ms	31.47 ms	94.4%
hpunch	0.44 ms	14.76 ms	15.20 ms	90.7%	2.19 ms	33.12 ms	35.31 ms	93.6%
iron	0.39 ms	11.82 ms	12.21 ms	91.9%	1.67 ms	19.38 ms	21.05 ms	92.7%
phone	0.40 ms	13.93 ms	14.33 ms	89.8%	1.97 ms	29.44 ms	31.41 ms	91.0%
Average	0.35 ms	11.60 ms	11.96 ms	92.9%	1.66 ms	23.92 ms	25.58 ms	94.4%
	20 Tree				50 Trees			
	T_Tree	T_Valid	T_Total	Acc.	T_Tree	T_Valid	T_Total	Acc.
ape	3.91 ms	16.42 ms	20.33 ms	97.2%	21.63 ms	28.94 ms	50.57 ms	97.2%
bvise	8.51 ms	33.51 ms	42.02 ms	93.1%	43.49 ms	38.66 ms	82.15 ms	93.2%
cam	8.15 ms	31.59 ms	39.74 ms	95.0%	42.03 ms	36.35 ms	78.38 ms	95.1%
can	8.73 ms	34.25 ms	42.98 ms	92.1%	44.73 ms	36.29 ms	81.02 ms	92.0%
cat	4.51 ms	19.33 ms	23.84 ms	95.5%	22.70 ms	29.84 ms	52.54 ms	95.5%
driller	9.73 ms	35.89 ms	45.62 ms	96.0%	49.29 ms	43.13 ms	92.42 ms	96.0%
duck	5.52 ms	21.63 ms	27.15 ms	94.7%	27.21 ms	32.70 ms	59.91 ms	94.7%
eggbox	5.99 ms	25.22 ms	31.21 ms	99.1%	30.89 ms	34.81 ms	65.70 ms	99.1%
glue	6.65 ms	34.89 ms	41.54 ms	93.9%	32.85 ms	44.17 ms	77.02 ms	94.4%
hpunch	8.65 ms	36.58 ms	45.23 ms	94.0%	42.93 ms	45.19 ms	88.12 ms	94.0%
iron	7.70 ms	24.20 ms	31.90 ms	93.6%	38.19 ms	33.63 ms	71.82 ms	93.6%
phone	7.90 ms	33.51 ms	41.41 ms	92.5%	37.06 ms	40.80 ms	77.86 ms	92.5%
Average	7.16 ms	28.92 ms	36.08 ms	94.7%	36.08 ms	37.04 ms	73.12 ms	94.7%

4.3. METHOD

Table 4.2: Accuracy and average time per frame for multiple object (5 trees) and its comparison with state-of-the-art approaches.

	5 Objects		13 Objects	
	T_Total	Acc.	T_Total	Acc.
ape	20.01 ms	97.3%	25.53 ms	97.4%
bvise	50.30 ms	94.4%	60.11 ms	93.4%
cam	59.81 ms	94.7%	64.47 ms	94.0%
can	53.54 ms	93.3%	66.98 ms	93.1%
cat	34.09 ms	95.2%	45.22 ms	96.0%
driller	66.40 ms	96.4%	78.94 ms	95.8%
duck	29.19 ms	95.5%	42.23 ms	96.2%
eggbox	34.50 ms	98.8%	47.98 ms	98.6%
glue	52.19 ms	94.7%	65.99 ms	94.6%
hpunch	56.32 ms	95.2%	74.56 ms	95.3%
iron	32.13 ms	93.2%	44.72 ms	93.6%
phone	54.71 ms	93.3%	56.10 ms	93.3%
Average	45.27 ms	95.2%	56.07 ms	95.1%
Hashmod	131 ms	95.5%	184 ms	95.1%
DTT-3D	107 ms	97.2%	239 ms	97.2%
LineMOD	427 ms	96.6%	1197 ms	96.6%

Table 4.3: Accuracy and average time per frame for self-comparison, evaluated on 13 objects. Each listed improvements lead to significant increases in performance. Baseline: kd-tree with Randomised decision forest (RF) by [Muja and Lowe, 2014]; improv. 1: Fuzzy split function (FZ); improv. 2: randomised ternary tree with extra rejector nodes (RN); improv 3.: Breath-first search in leaf validation (LV).

	Time	Avg. acc.
Random Forest (100 trees)	963.43 ms	73.58%
RF + RN	133.12 ms	67.35%
RF + RN + FZ (5 trees)	201.32 ms	95.3%
RF + FZ + RN + LV	45.27 ms	95.1%

4.3.5 Evaluation

Experiments are conducted on LineMOD ACCV12 dataset [Hinterstoisser et al., 2012b], which consists of 13 object (we omitted 2 objects since proper 3D models were missing) CAD models and 15 testing sequences for object detection and 6D pose estimation. Each sequence has 1100 images covering different viewpoints, distances in clutter. As we do not exploit the temporal information in this work, each image is processed individually. It is hard to measure the error in 6-DoF parameter space. Therefore it is typical to measure the reconstruction error instead, defined as below,

$$\epsilon = \text{avg}_{x \in \mathcal{M}} \| (\hat{r}x + \hat{c}) - (rx + c) \|, \quad (4.14)$$

where (\hat{r}, \hat{c}) is groundtruth pose and (r, c) is the pose annotated with the retrieved template. Following [Hinterstoisser et al., 2012b], we also use the criterion $k_m d_{\mathcal{M}} > \epsilon$ to decide whether an object instance is detected, where $d_{\mathcal{M}}$ is the diameter of object \mathcal{M} , and k_m is a coefficient set to $0.1m$, same as in [Hinterstoisser et al., 2012b]. All experiments are conducted on a single 2.8 GHz Intel Core i7. A pyramid scheme is applied in a similar way to [Hinterstoisser et al., 2012b].

In general, our pipeline achieves sublinear time complexity and comparable high accuracy as shown in Table 4.1. Despite the detect rate of our approach being marginally worse than state-of-the-art approaches, we are at least two times faster than the fastest DTT-3D [Rios-Cabrera and Tuytelaars, 2013]. Since our approach is tree-based, it is also scalable to more objects. Table 4.2 shows that we significantly outperform state-of-the-art approaches in speed with more objects. Additionally, both tables show that our approach works favourably on simpler objects, because the training templates share more similar features so the background is more likely to be rejected before the expensive validation stage.

In Figure 4.4, we illustrate the effectiveness of our proposed preemptive background rejector. Depending on the object complexity and scene, up to 90% to 97% background locations can be filtered out before the validation stage with a very high recall rate. Since textureless objects are generally simple in shape and colour, their rendered templates are likely to share particular features that can easily rule out background clutters. The parameter analysis for decision tree depth and fuzzy factor is shown in Figure 4.5. The runtime is inversely proportional to the tree depth and exponentially grows with the fuzzy factor ζ . With regard to tree depth, the accuracy drops drastically due to error accumulation when leaf nodes contain too few templates. The bottom right figure also shows the necessity of fuzzy split to reduce the error accumulation in decision forest.

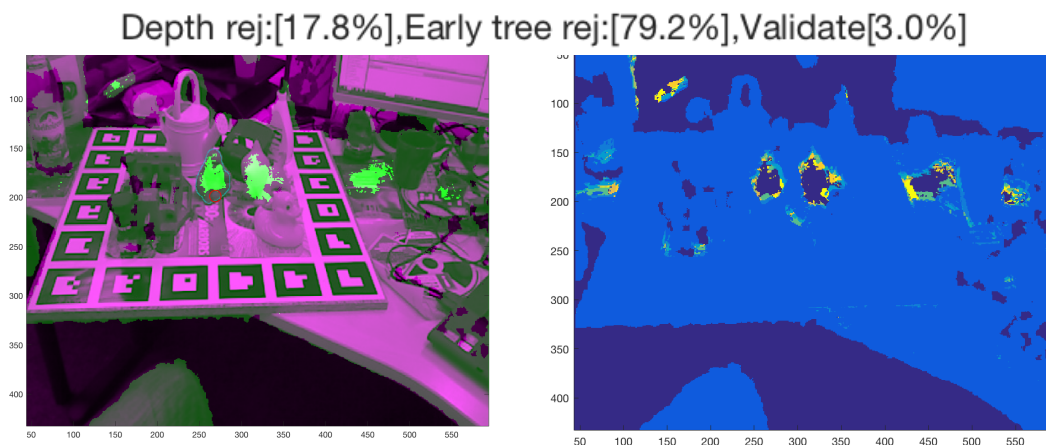


Figure 4.4: In this sample frame, with our proposed preemptive background rejector up to 97% negative bounding boxes are filtered before reaching the forest leaf nodes. The green region in left image indicates the locations that enter the validation stage; the right image shows the tree depth when background locations are rejected, dark blue indicates the region is either rejected due to out-of-depth-range or pass the validation stage, light blue to yellow indicates whether the locations are rejected early or late in the forest. As nearer to the ground truth or ambiguous objects the location is more likely to not be rejected earlier.

Since majority of negative proposal locations are rejected at the first node in the decision tree, in Table 4.1 we show that the time cost on testing tree itself (T_{Tree}) is neg-

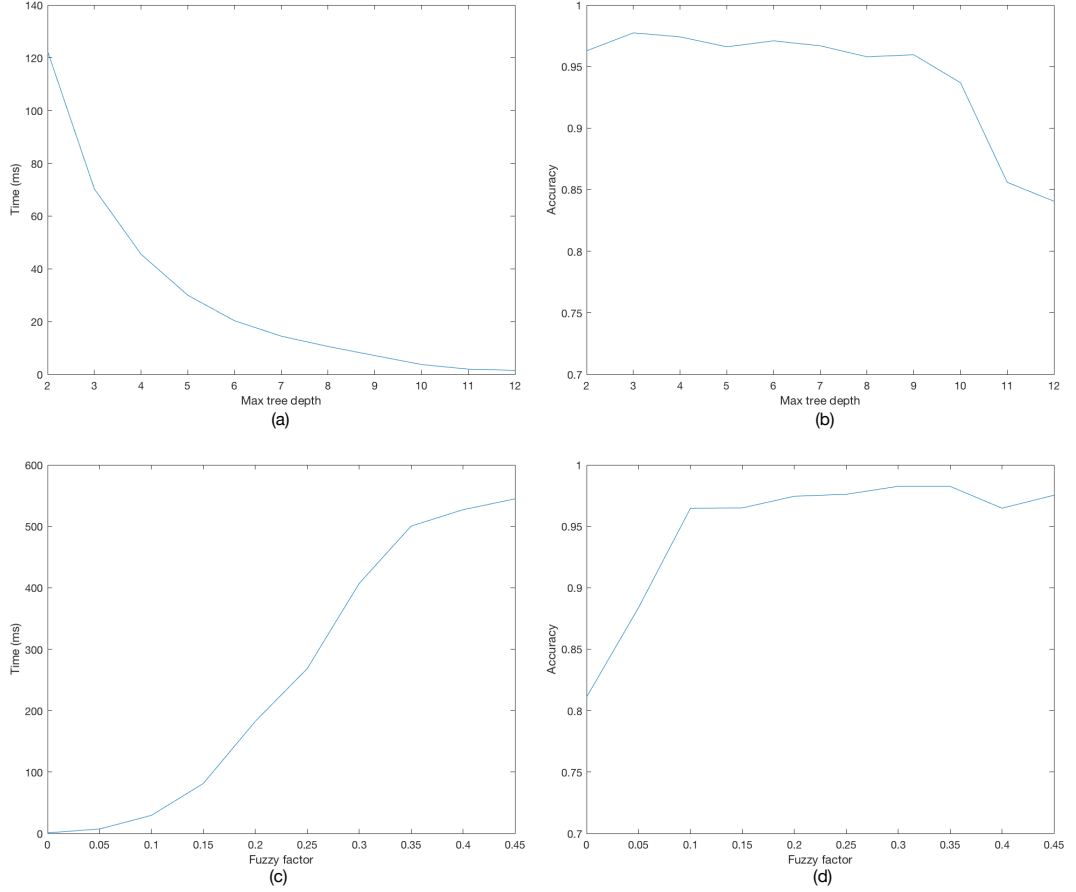


Figure 4.5: Top: max decision tree depth versus runtime and accuracy. Bottom: fuzzy factor ζ versus runtime and accuracy.

ligible compare to the validation time (T_{Valid}). The performance is further boosted with multiple trees with random permutations. Here, our proposed approach shows another advantage with random forest framework. The time cost for additional trees has sublinear growth, as the leaf nodes in each tree are concatenated to remove duplications before entering the validation stage. The result shows that the accuracy is increased by 1.5% with 5 trees but only approximately 2 times slower.

For the adaption of fast breadth-first leaf validation, the time taken is reduced up to 5 times without loss of accuracy. Note that in the practice, this indexing approach is less efficient if the template set in each leaf node is too small. In our case, we set our

maximum tree depth to be 8, and 9 for multiple objects.

Finally, our approach achieves sublinear complexity from the binary tree structure and significantly outperform the state-of-the-art approaches in terms of processing speed.

4.3.6 Summary

In this chapter we study the impact of various outlier removal techniques on classical randomised decision forest framework to solve 3D object detection and pose estimation problem. Along the method we have made a number of modifications on a classical randomised decision forest framework.

We first adapt a ‘fuzzy’ split function to deal with noisy, over quantised LineMOD template features and insufficient samples. A fuzzy factor is proposed to control the tolerance of ambiguous feature within each split node. During decision forest growth, training samples that near to the partition subspace are duplicated to both child nodes. Similarly, during the inferencing, any query sample that is too close to the split subspace will be traversed to both child nodes.

Next we augment the split function with a preemptive background rejector to handle background noises and a nearest neighbour pairwise matching at the end. This extends the original binary test on each split node into a ternary test that aims to early terminate the inferencing if testing sample falls into any extra background node. While the parameters of each split node is learnt by maximising information gain, background rejector is in contract learnt by minimising the information gain. The idea behind is to find common features shared amongst the training subsets, i.e., feature dimensions that have the the lowest entropy, and by assuming negative samples distribute equally in the feature space, we can maximise our possibility to detect and remove background samples at the earliest.

Finally we adopt breadth-first searching technique for leaf validation, inspired by the preemptive RANSAC [Nistér, 2005a]. Unlike classical decision forest, in our method, a testing sample can be traversed to multiple leaves due to the fuzzy split rule, and each leaf contains multiple samples. Instead of expensive pairwise matching, we iteratively select a random feature subspace and pairwise measure the feature distances only within this subspace, then remove sample candidates that do not satisfy the pass condition.

The result shows that we significantly outperform the state-of-the-art template matching methods in terms of speed while maintaining a reasonable precision. For future work, it is intriguing to further develop the background rejection scheme into a more up-to-date framework, such as deep neural network for saving intensive GPU workload.

4.3.7 Generalisation

In the previous chapter, we discuss the benefits of finding and exploiting suitable data attributes from heuristic or human knowledge, to apply outlier removal techniques as preprocessor to improve the machine learning system performance. It will general lead to less training data requirement, less complexity of model, hence and a more efficient system.

In this chapter, we focus on accelerating outlier removal process with a novel early outlier rejection technique. This approach merges the merits of cascading architecture, template representation and classical randomised decision forest framework. To compensate the low detection rate of each split node, we also adopt a fuzzy split function to ensure the detection rate of each split node, with little sacrifices of algorithm complexity.

This approach can be generalised to any machine learning task that has very low positive rate, such as for typical face (or object) detection problem, on average only 0.01% of all sub-windows are positive [Viola et al., 2001]. This assumption is especially true in most real-world applications.

Moreover, this approach is limited only to random forest framework, it has potential to be implemented to any directed-acyclic-graph-(DAG)-based classifier, such as deep convolutional neural network (CNN). It is widely known that typical CNN forward propagation consumes very high computational power and require specialised hardware (GPUs) with high power consumption, often around 250W per card. This motivates the implementation of early termination operator to CNN framework.

4.3.8 Next Chapter

Next chapter presents a case study, we discover a phenomenon that in industrial shape fitting tasks, the extracted edge point outliers are likely to appear in group, thus we propose a simple heuristic outlier removal method to deal with grouped outliers. We show that despite outliers are theoretically unpredictable, in real-world industrial applications they likely to follow a specific pattern, depending on the environment, so that it can be leveraged to assist outlier removal method.

5

CHAPTER

A GROUPED-OUTLIER-AWARE REGISTRATION-BASED METHOD FOR ROBUST ELLIPSE FITTING

CONTENTS

5.1 Overview	88
5.2 Problem Setting and Preliminaries	90
5.3 Proposed Approach	93
5.4 Evaluation	97
5.5 Summary	100

5.1 Overview

Ellipse fitting is one of the fundamental problems in computer vision and robotic tasks. It is required as preprocessing modules in many high level computer vision applications such as textureless object recognition and shape alignment. In this chapter, we study the impact of an outlier removal on a classical industrial ellipse fitting task, and propose a real-time outlier removal solution to deal with a special type of outlier that commonly exists in real-life tasks, which we call them ‘grouped’ outliers – a set of outliers that are contaminated in a similar manner. This work has been done during my internship in OMRON, Japan. ¹.

Comparing to other chapters, this chapter presents a case study of outlier removal method for assisting real-life industrial shape fitting task, as the proposed method is simpler and more heuristic compare to previous works. However, in real-life problem we often have to deal with situations that its classical solution does not perform optimally as some conditions cannot be satisfied.

For instance, the literature of ellipse fitting methods mostly assume the only error on the location of shape edge points are Gaussian distributed, thus efforts have been made to approach the theoretical accuracy bound, or KCR lower bound [Chernov and Lesort, 2004]. When other types of noise or unknown outliers appear, the noise model no longer suits thus leading to a suboptimal fitting result.

Therefore in industrial shape fitting applications, outlier removal is applied to deal with unpredictable noises. In the real-world scenario, image or extracted edges are usually contaminated heavily when under partial occlusion, specular highlight, deformation, shading or other environmental nuisances. In such cases, robust fitting algorithm like random sample consensus (RANSAC) [Fischler and Bolles, 1981a] is gener-

¹For the contribution, Dr. Ijiri had helped the idea discussion, Mr. Hattori had helped the dataset collection

ally applied to eliminate outliers. However, when the inlier rate ϵ is low, RANSAC soon becomes infeasible due to the possibility of finding at least one correct ellipse model is $p = 1 - (1 - \epsilon^5)^K$, where K is the number of iterations.

For using outlier removal method in shape fitting task, a recent work by [Yu et al., 2010] proposed a proximity-based outlier detection algorithm to effectively remove the isolated outliers and outlier clusters by constructing a proximity graph. However, the adjacency matrix is too expensive to compute if the data point set is large and moreover, some parameters need to be tuned carefully in order to achieve a good clustering result. Also, the proximity-based method fails if the outliers are connected smoothly with the inliers. Another work [Prasad et al., 2013] presents an accurate, non-iterative method based on the geometric distance between a data point and an ellipse, but the method is still not efficient enough to be used in a light weight real-time shape fitting method.

From practical observations on sensor images, we realised that in industrial shape fitting task, outliers mostly do not appear completely in a random manner. Instead they appear in group and somewhat follow a particular pattern. One example is shown in Figure 5.1. How outlier groups under environmental nuisances is hard to be learnt by machine learning techniques since ‘the pattern’ varies diversely from case to case. However, from observations we know that the contaminated shape edge points are mostly clustered on some small area of the image. Thus if we can find a efficient way to group the edge points into pure inlier and outlier subsets, the computational complexity of classical outlier removal methods can be drastically reduced.

In this work, we demonstrate how edge points can be effectively grouped into short contours to reduce the computational cost, and further show how to eliminate the outlier contours in a breadth-first manner. Our contribution is four-fold. First we introduce a split-and-merge trick to cluster data points into subsets that contain either pure inliers or outliers. Second we propose a breadth-first strategy for searching the outlier contours through the combination of subsets. Third, we speed up the searching pro-

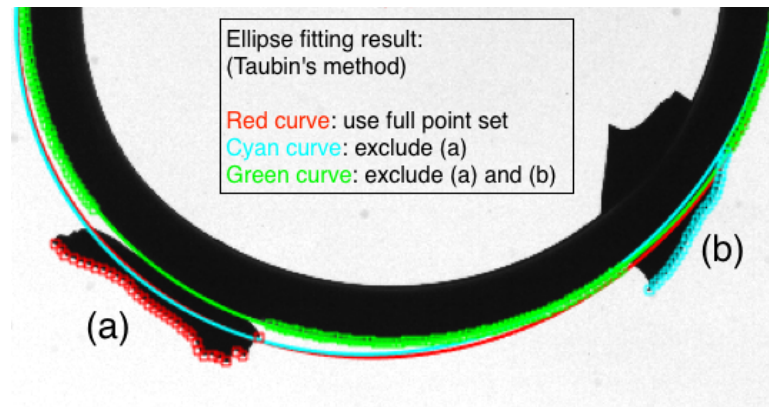


Figure 5.1: A challenging case for proximity-based outlier removal method. Type (a) outliers cannot be filtered by simple proximity (e.g. k-Nearest Neighbour) check. Type (b) is even more difficult since they are connected with the inlier contour.

cess by using the smallest generalised eigenvalue (which is a by-product of the chosen algebraic fitting algorithm) to approximate the point-to-curve projection, and then use the algebraic fitting solution as an initial guess for geometric fitting algorithm to alleviate measurement noises. Finally, we proposed both realistic and synthetic datasets for evaluation.

5.2 Problem Setting and Preliminaries

Given a point set, the objective of ellipse fitting is to find a geometric parameter set that minimises the sum of inlier-points-to-ellipse-curve projection distance. In this work, we assume that the edge points are already extracted and follow a cyclic order. Specifically, given an approximated ellipse center point, the edge points are collected circularly by finding the maximal gradient change along all radii. In consequence, the arrange of the point sequence is naturally known. This setting commonly exists in many industrial applications, such like shape alignment between mechanical components.

In the following subsections, we will briefly introduce two approaches for shape

fitting: algebraic (Least-Square-based) and geometric (Maximum-Likelihood-based) method.

5.2.1 Algebraic Fitting

Any ellipse can be represented by a second order polynomial $F(\mathbf{u}, \mathbf{x}) = \mathbf{u} \cdot \mathbf{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0$, subject to $b^2 < 2ac$. Our goal here is to estimate the parameter set $\mathbf{u} = (a, b, c, d, e, f)$ from a given point set $\mathbf{x}_i = (x_i, y_i)$ such that the sum of algebraic distance $\sum_{i=1}^N |F(\mathbf{u}, \mathbf{x}_i)|_2$ is minimised.

This problem is generally tackled with linear least square solvers as in several seminal approaches [Taubin, 1991, Fitzgibbon et al., 1999, Kanatani and Rangarajan, 2010]. Since the aim of this chapter is to study how outlier removal techniques improves the ellipse fitting method, we use the classic Taubin's method [Taubin, 1991] as our off-the-shelf algebraic ellipse fitting algorithm. Also, according our empirical experiments, Taubin's method remains one of the most accurate and robust methods given its efficiency despite it has been proposed for decades. Since this method is designed for general conic fitting, it might return conics other than the ellipse (e.g. hyperbola curve), causing incorrect fitting result.

In Taubin's method, the solution is given by solving the generalised eigenvalue problem:

$$\mathbf{M}\mathbf{u} = \lambda\mathbf{N}\mathbf{u}$$

$$\text{where } \mathbf{M} = \frac{1}{N} \sum_{n=1}^N \tilde{\xi}_n^T \tilde{\xi}_n,$$

$$\tilde{\xi} = (x^2 \ xy \ y^2 \ f_0x \ f_0y \ f_0^2)^T,$$

$$\mathbf{N} = \frac{4}{N} \sum_{n=1}^N \begin{pmatrix} x_n^2 & x_n y_n & 0 & f_0 x_n & 0 & 0 \\ x_n y_n & x_n^2 + y_n^2 & x_n y_n & f_0 y_n & f_0 x_n & 0 \\ 0 & x_n y_n & y_n^2 & 0 & f_0 y_n & 0 \\ f_0 x_n & f_0 y_n & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 x_n & f_0 y_n & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The solution \mathbf{u} is given by the the smallest generalised eigenvalue λ .

5.2.2 Geometric Fitting

Geometric parameters of ellipse consist of 4 elements: center position, length of major axis, minor axis and angle of tilt. Unlike algebraic methods, the linear condition has no longer held for solving the least square problem with geometric parameters. Instead, iterative optimisation methods are used to find the local optimal solution given an initial guess.

Geometric fitting, i.e. , Maximum-Likelihood-based method is generally regarded as one of the most precise fitting algorithms. They do not suffer from scale indeterminacy as in the algebraic methods and are likely to achieve the local optimal solution. A detailed analysis is given in the work of [Kanatani and Rangarajan, 2010].

The main drawback of geometric fitting algorithms is their high computational cost due to iterative optimisation process. With the existence of outliers or high measurement noise, they also hard to converge. The initial guess is crucial as well to achieve the global optimal solution.

In this work, we apply geometric fitting as the final refinement to alleviate the measurement noise while keeping the pipeline fast by using an algebraic solution as initial guess.

5.3 Proposed Approach

The proposed full pipeline consists of three stages:

- Clustering data points based on proximity so that each subset is likely to contain only inliers or outliers edge points.
- Searching through combinations of subsets to minimise algebraic fitting distance until convergence.
- Refining the algebraic solution by the geometric fitting algorithm.

5.3.1 Proximity-based Point Clustering

Generally, a partial or full adjacent matrix needs to be calculated to determine the connections between edge points. This process roughly has computational complexity $O(N^2)$ with respect to the number of data points, and an adequate connection radius is also needed to be carefully chosen. The edge point detector we used in this work naturally provides the connectivity between the points, which simplifies the clustering process considerably.

The point set is ordered by the edge point detector such that \mathbf{x}_n and \mathbf{x}_{n+1} are next to each other. A point set is split at a point that Euclidean distance to its neighbour points d_i greater than $t * \text{median}_i\{d_i\}$, where t is the distance ratio threshold. If the measurement noise is too high, t should be set to a larger number to prevent over-segmentation. In this step, the subset is discarded as isolated outlier if it contains less than τ points.

To deal with type (b) outliers in Figure 5.1, each subset is uniformly split if they contain more than a certain number of points. This number is determined by a pre-

Algorithm 1: Proximity-based Point Clustering

```

Input:  $C_{full} = \{\mathbf{x}_i\}_{i=1}^N$ 
Initialisation:  $t, \tau, D, C_1 = \{\}$ ;
for  $i \leftarrow 1$  to  $N$  do
     $C_k = C_k \cup \mathbf{x}_i$ ;
     $d_i = \|\mathbf{x}_i - \mathbf{x}_{i+1}\|_2$ ; /*  $i+1 = 1$  if  $i = N$  */
    if  $d_i > t * \text{median}\{d_i\}$  then
         $k = k + 1$ ;  $C_k = \{\}$ ;
    end
end
Delete any set  $C_k$  that has cardinality  $|C_k| < \tau$ 
for  $k \leftarrow 1$  to  $K$  do
    if  $|C_k| > \frac{2N}{D}$  then
        Uniformly split  $C_k$  into  $\lfloor \frac{|C_k|}{r} \rfloor$  sets;
        Replace  $C_k$  by these sets;
    end
    else if  $|C_k| + |C_{k+1}| < \frac{N}{D}$  then
         $C_k = C_k \cup C_{k+1}$ ; /* Delete set  $C_{k+1}$  */
    end
end
return  $\{C_k\}_{k=1}^K$ 

```

defined expected subset cardinality D . Larger D allows for a finer segmentation of edge points, but induces higher risk of getting stuck in local optimum after the later searching stage, and also sacrifices the processing speed.

In the last step, neighbouring subsets are merged if their sum of cardinality is sufficiently small. In the end, the whole point set should be split into similar sizes and each subset is likely to contain only inliers or outliers. The pseudo code of this section is shown in Algorithm 1.

5.3.2 Breadth-First Searching

As shown in Figure 5.2, the fitting trials are performed between the combinations of subsets only. The total number of possible combinations $\{C\}_{n=1}^D$ is $\sum_{n=0}^D \frac{D!}{n!(D-n)!}$, could be still large in number.

For each test, Taubin's fitting method [Taubin, 1991] is used to sidestep the expen-

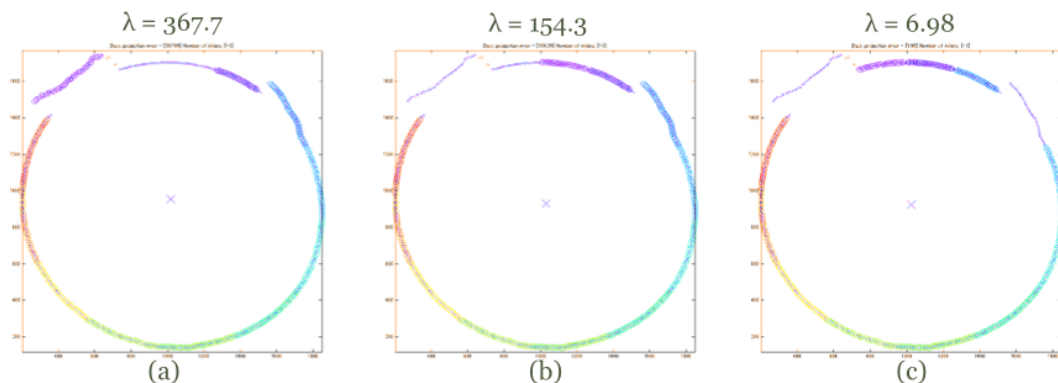


Figure 5.2: This figure shows an example that the value of the smallest generalised eigenvalue λ decreasing drastically after excluding the outlier subsets. The point set contains two ‘outlier’ subsets, (a), (b) and (c) show three cases during the iteration, while (a) does not exclude any outlier subset, (b) excludes one of them and (c) excludes all.

sive geometric projection. We further speed up the process by replacing algebraic distance to the smallest generalised eigenvalue λ when solving the linear least square system, since they are comonotonic.

Here we propose a searching scheme that efficiently searches and excludes the outlier subsets in a breadth-first manner. To minimise the energy while maximising the number of possible inliers, combinations of sets are tested by excluding one subset in each iteration. The searching process converges if excluding any subset does not reduce the energy upto certain rate σ . To prevent local optimal solution, number of S candidate sets with lowest energy are stored at each iteration, and each will be appended with other S candidates in the next iteration. In case there are much more outliers than average subset size so that removing any subset does not reduce the energy, we introduce an error threshold η as another stop condition. The pseudo code of this section is provided in Algorithm 2.

Algorithm 2: Preemptive searching

```

Input:  $\{C_k\}_{k=1}^K$ 
Initialisation:  $S, \sigma, \eta, \{O_s = \emptyset\}_{s=1}^S$ ;
for  $m \leftarrow 1$  to  $K$  do
     $C_{test} = \bigcup_{k \in K, k \neq m} \{C_k\}$ ;
     $[\lambda_{m,1}, \mathbf{u}_{m,1}] = \text{TaubinFitting}(C_{test})$ ;
end
 $O_s = \{m\}, E_s = \lambda_m$  for  $s^{th}$  least  $\lambda_m$ ;
for  $l \leftarrow K - 2$  to  $1$  do
    for  $s \leftarrow 1$  to  $|\{O_s\}|$  do
        for  $m \leftarrow 1$  to  $K, m \notin O_s$  do
             $C_{test} = \bigcup_{k \in K, k \neq m} \{C_k\}$ ;
             $[\lambda_{m,s}, \mathbf{u}_{m,s}] = \text{TaubinFitting}(C_{test})$ ;
        end
    end
    if  $(E_1 > \sigma * \min \{\lambda_{m,s}\}) \& (E_1 < \eta)$  then
        return  $\mathbf{u}_{\underset{(m,s)}{\text{argmin}} \{\lambda_{m,s}\}}$ 
    end
    else
         $S \leftarrow l$  if  $S > l$ ;
        for  $s \leftarrow 1$  to  $|\{O_s\}|$  do
            replace  $\{O_s\}$  by  $\{O_s \cup \{m\}\}$  for  $S$  least  $\lambda_{m,s}$ ;
        end
        update  $E_s$ ;
    end
end

```

5.3.3 Refinement

Experimentally we found a refinement step to be crucial to reducing the measurement noise from the edge point detector. Given the inlier point set, we convert the algebraic parameters \mathbf{u} to geometric parameters H and apply a nonlinear optimiser as below to further minimise the point-to-curve (orthogonal to the ellipse tangent) projection error. The detail for calculating projection distance can be found in [Eberly, 1998].

$$H^* = \underset{H}{\text{argmin}} \|\text{proj}(H, C)\|_2$$

Since the initial geometric parameters were already estimated from the previous stages, they are close enough to the optimal solution. Thus, the efficiency is maintained

despite the algorithm itself being expensive, because it converges within few steps in most cases.

5.4 Evaluation

Our method is evaluated by both synthetic¹ and realistic datasets which consist of ellipse edge points with different types of contamination. Both datasets are collected using the same off-the-shelf edge point detection method introduced in ‘Preliminaries’ section.

For synthetic data, we have rendered 3 scenes (2 with occlusions and 1 with background clutter) using Blender (an open-sourced 3D computer graphic software) at 960×540 resolution. The scenes consist of an arbitrary sized ellipse and several distractors, either colour-filled rectangle or circles. Each scene is animated and contains 100 frames, each frame consists of 100 extracted ellipse edge points. The ground truth ellipse centers are fixed to the image center. To simulate the measurement noise from camera, we augmented Gaussian noise ($\sigma = 3$ pixels) to the data points. For private realistic dataset, each image sample contains an industrial object with the shape of ellipse. Each object is then manually fit with an ellipse curve so that center location and edge points can be extracted. We use the following parameter setting for our method in all experiments: $t = 2$, $\tau = 5$, $D = 12$, $S = D/2$, $\eta = 5$ and $\sigma = 1.05$. The parameters are also chosen with a grid search, however, the experiments on realistic and synthetic dataset suggests that this parameter set adapts a wide range of inlier rate and outlier type. It is believed that the parameters set can be generalised to other situation without retuning. The reason for exposing all parameters for tuning is that for industrial applications, the environment is often consistent, such as a fixed camera on

¹<http://bit.ly/1Dvs0id>

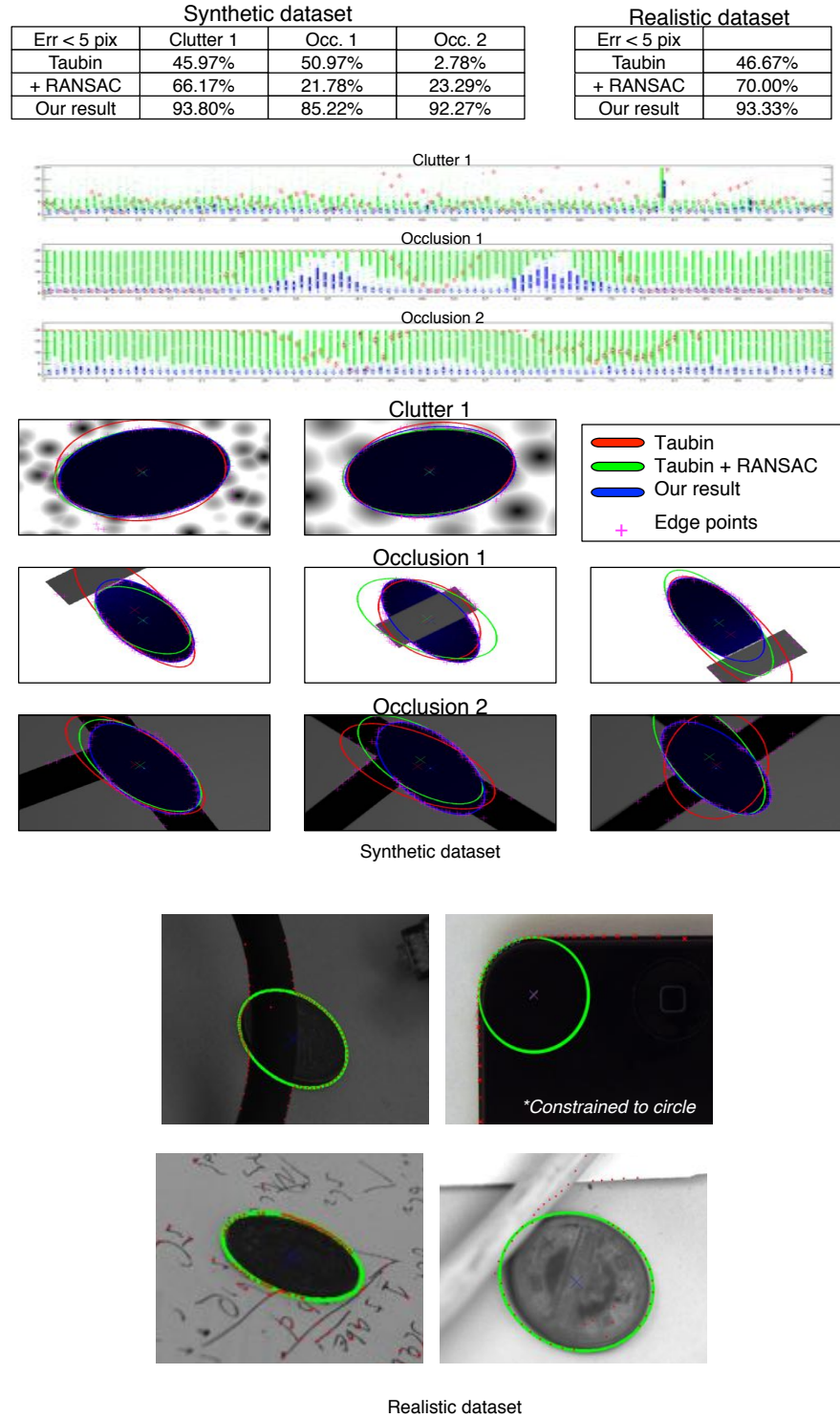


Figure 5.3: Evaluation result of 3 methods on synthetic and realistic datasets (best view in color).

an assembly line. Therefore users demand an extremely optimal parameter set by a precise parameter tuning, involving all tuneable parameters.

We compare our method with Taubin’s method with/without RANSAC, and summarise the evaluation result in Figure 5.3. The boxplot reflects the statistic of center errors (capped at 20 pixel) from 100 runs of each method on every video frames. The table on the top left corner shows the success rate (center error less than 5 pixels) of the methods on each video. Note that for synthetic dataset, Gaussian noise is augmented to all data points in each run to simulate the noise from realistic optical sensors. Overall, our approach achieves superior accuracy than the baselines.

For Taubin’s method itself, almost all fitting estimations are distorted by outliers due to the nature of least square solvers, such as in the dataset ‘Occlusion 2’ shown in Figure 5.3. However, RANSAC does not boost the accuracy as significantly as expected either. The main reason is twofold. Firstly the parameters, e.g. , inlier ratio and distance threshold, are employed empirically as stop criteria. Such settings are not optimal to all situations, therefore leading to an overall poor performance. Another is due to the high measurement noise within all data points. Since RANSAC picks hypotheses from random minimal samples, the impact from noises is drastically amplified compared to estimating from all inlier samples, which also explains the motivation of our method. There are several extended RANSAC-like methods that have been proposed to deal with such a problem, but they either are expensive (e.g. , pre-emptive RANSAC [Nistér, 2005b]) for ellipse fitting problem, or suffer from poor initial estimations generated from minimal samples(e.g. locally optimised RANSAC [Chum et al., 2003]).

Our method achieves less than 20ms runtime on single core CPU with up to 360 data points, which meets the time requirement as a pre-processing module for many higher level real-time applications. With code optimisation and parallel processing, the whole pipeline will further speed up.

5.5 Summary

This chapter presented a simple outlier removal technique to solve real-time ellipses fitting under high outlier rate. We used an off-the-shelf ellipse fitting method in the experiment, however it can be easily replaced to any shape fitting method. We demonstrated how shape edge points can be clustered based on their proximity, and how outliers can be filtered in a ‘preemptive’ manner. Our method has shown to be effective on the ‘grouped’ outliers due to common industrial environmental nuisances such as partial occlusion, partial deformation and specular highlight. If the pattern of outliers do not follow the assumption, the performance is comparable to the baseline. The runtime of the overall method is below 20ms in average on a moderate single CPU core, for up to 360 edge points, which is sufficient light weight to achieve real-time processing speed for many higher level applications.

5.5.1 Generalisation

The insight of the proposed method is to split the data into subsets that contain pure inliers or outliers, achieved by either simple proximity check or more complex label propagation techniques. This approach can simplify a behavioural or collective outlier removal problem into a contextual or individual outlier removal problem.

Although in this case study we cluster the points using a simple proximity check, in general the clustering process should subject to the pattern of inlier and outlier sets. One practical example would be removing false detection in an optical character recognition system (OCR) for industrial use. If it is known that the characters are mostly written horizontally or vertically, clustering the character detection proposals with line fitting algorithm would remove most of false proposals with a very low cost.

Moreover, clustering strategies can be learnt through machine learning techniques. There are many recent works on learning models to represent the relationship between common distinctive parts in the data, such as [Zhang et al., 2015] proposed a hierarchical And-Or graph model to represent the object classes into connections between common unannotated attributions from object images. The widely applied convolutional neural network models also capture these relationships intrinsically. These models can actually be leveraged for removing outlier groups that do not follow any known pattern from the dataset.

In a more general aspect, this chapter also demonstrates the importance of analysing and then exploiting the patterns within outlier sets, despite outliers are theoretical cannot be learnt. In real-world problems, since an outlier is always true signal that generated by anomalous causes, in some cases seeking how outliers are generated cost much less than learning the inlier-outlier boundary. These problems are also often referred as anomaly detection or novelty detection.

In this study case, which is under an industrial background, we have found that the outliers of extracted object edge points are mostly generated from the occlusion or due to incomplete object part. This fact causes the outliers are clustered and contextual, so that a method for dealing grouped outliers are proposed to remove this specific type of outliers. For other real-world machine learning tasks, this approach can be applied as an extra refine step of generic outlier removal method to deal with complex outliers. One example would be persistent false positive that cannot be dealt by the classifier, such as under surveillance camera, a plant is kept being detected as a person. In such case, we may manually label the plant to train an extra classifier or apply novelty detection to learn a ‘objects-that-similar-to-human’ outlier filter, over the surveillance history. There are plenty related works on novelty detection, but not much on the combination of novelty detector and outlier removal.

CHAPTER

6

CONCLUSION AND FUTURE WORKS

CONTENTS

6.1 Conclusions	103
6.2 Future Works	105

6.1 Conclusions

In this thesis, we investigate the problem of outlier removal in various computer vision tasks. In this chapter, we summarise the thesis contributions and discuss future directions.

In Chapter 1, we introduce the problem definition and application examples of object recognition and pose estimation, and the importance of outlier removal algorithm in these tasks. Then, we discuss the challenges within, and hence the motivations for

further developments. Finally, we list the thesis structure and summarise the contributions. Chapter 2 provides a general review of literatures of outlier removal methods from a machine learning point of view. In this chapter, we categorise existing outlier removal techniques into distance-based, learning-based and registration-based. In each category, we provide a brief explanation and several seminal related works.

Chapter 3 studies how spatial and temporal cues can be exploited for outlier removal under modern rigid object recognition frameworks (Chapter 3). In this chapter we conduct a comparative study and then propose a new video dataset with 33 less textured sculptures in cluttered museum scenes. We implement several state-of-the-art object recognition frameworks and their extensions, including patch-based local descriptor matching with geometric constraint (2D-to-2D, 2D-to-3D and 3D-to-3D), bag-of-words-like global image descriptor, set-to-set kernel principle angle approach and so on. From the evaluation result, we draw conclusion that exploiting 3D geometric cue from the nature of rigid objects can vastly improve image classification accuracy, and also it removes a large portion of negative local image patches, hence vastly improves the matching speed. Another temporal spatial cue within the adjacent frames in the video can be leveraged to remove noise and compress the feature set.

In chapter 4 we propose a learning-based outlier removal approach for real-time 3D object pose estimation. We show that it is viable and beneficial to implement various outlier removal techniques into the randomised decision forest framework. This idea of ‘early termination’ presents another way of using outlier removal methods to improve state-of-the-art machine learning frameworks.

In chapter 5 we present a simple outlier removal technique to solve real-time ellipses fitting under high outlier rate. Our method has shown to be effective on the ‘grouped’ outliers due to common industrial environmental nuisances such as partial occlusion, partial deformation and specular highlight.

6.2 Future Works

The future works based on this thesis are summarised as below:

- Chapter 3: apply meta-learning techniques to learn the characteristic of given datasets (inliers), hence enhance the performance of outlier removal or machine learning systems in general.
- Chapter 4: extend early-termination of outlier query to other framework for less computation cost, such as deep convolutional neural network framework (CNN).
- Chapter 5: apply anomaly detection or novelty detection algorithms to learn the characteristic of outliers.

6.2.1 Learn characteristic of inliers via meta-learning technique

To recognise an object in the real world, the 3D geometric cue is known to be straight forward to human being: a single glance of a rigid object is almost enough to recognise it from any other viewpoints. However, this is in contrast to the success of most modern general classification/detection frameworks, which relies on the 'Big Data' and powerful processors (GPUs). The current up-to-date solution, as known as deep convolutional neural network (CNN), constructs a huge parameter space to fit any kind of data. Usually a well-trained CNN model consists of million float-point numbers as parameter. There are plenty of works trying to reduce the model complexity, such as transfer learning and knowledge distillation [Hinton et al., 2015], compressing the model using quantisation or pruning techniques, to make it possible to deploy heavy CNN models into embedded devices. That said, the CNN models still consume a large

amount of computational resources and work like black boxes that cannot easily leverage or capture semantics. On the other hand, achieving state-of-the-art performance often requires millions of labelled image data.

When solving real-world computer vision problems, these facts encourage us to not entirely rely on the data stream from sensors or huge models, but also exploit the attributes within them – the information extracted from human common knowledge and do not require intensive labelling effort. For instance, sculptures do not deform so that their 3D geometry can be easily used to validate their 2D images. Despite we may concatenate these attributes to the data labels, there is unfortunately no machine learning framework could train from all sort of information. However, we may use an ensemble of ‘micro-frameworks’ instead of a big universal framework, and how it composed should be problem specific. In our comparative study in chapter 3, we show that the best performance is achieved by using a hybrid method that combines local image descriptors matching, 2D-to-3D geometric validation and other outlier removal techniques.

The drawback of such approach is obvious, as handcrafting a pipeline of methods is not desirable in the world of machine learning. So one future work could be first split state-of-the-art frameworks into components, then training a machine learning model that learn the optimal combination given the dataset. Such thinking is proven to be effective under deep neural network framework, one example is known as NASNet [Zoph et al., 2018], where it learns the model architectures directly on the dataset of interest. It would be intriguing to generalise it to other machine learning frameworks. Outlier removal methods would also largely benefit from this approach as they are simple, effective but problem specific and sometimes hard to generalise to other tasks.

In a more general aspect, this also can be seen as a ‘meta-learning’ problem, where we seek the best machine learning approach based on the ‘meta-knowledge’ of a given dataset, i.e. the characteristics of the data. For video-based rigid object recog-

nition or pose estimation problem, the ‘meta-knowledge’ would include the 3D geometry and temporo-spatial information, which leads to a workflow that involves 3D-reconstruction, 2D-to-3D geometric validation and optical flow.

By now, the meta-knowledge of a given dataset is still mainly extracted manually with human common sense, it would be intriguing if the meta-knowledge can be learnt via machine learning methods as human do. In the literature, one approach is via Reinforcement Learning (RL), where the algorithm learns to accelerate reward intake by continually improving its own learning algorithm which is part of the self-referential policy ([Schmidhuber et al., 1997]). This is also closely related to few-shot learning, where a model is learnt while the data amount is very limited.

The challenge is obvious, we still expect a good meta-learning model to be able to generalise a given set of concepts to the new data sample. Also, how do different concepts embed to the model remains a difficult problem.

6.2.2 Early termination of outlier query in CNN framework

To meet the processing speed requirement of real-world applications, it is important to discard the background or uninterested query proposal as quickly as possible. In case of the most widely used machine learning recently, deep CNN, it should not be too difficult to implement intermediate outlier removal layers in order to save computation from background regions. Concretely, such as the zero-valued activations that arise from the rectified linear unit (ReLU) operator could be exploited to prevent unnecessary data transfers, or early termination of further processing.

Though, this approach imposes a greater challenge to the hardware architecture for fast CNN inference. Due to the optimisation of matrix operation in the processor, typical convolutional dataflows cannot be gated or stopped. One solution is the

redesign of CNN inference accelerator architecture, one recent work [Parashar et al., 2017] designed a zero-value-aware accelerator architecture and improves performance and energy efficiency by a factor of 2.7 and 2.3. Another possible direction would be send the intermediate data back and forth between CPU and accelerator, so that the computation on the sparse feature map can be easily implemented.

6.2.3 Learn characteristic of outliers via anomaly detection technique

Instead of learning characteristics of inlier as in Chapter 3, it is also possible to learn from outliers. To solve real-world problem, it is sometimes worth to model outliers under specific environment. The future work may include online learning of anomaly data for outlier removal, which would be valuable for many surveillance systems. There are many related works on anomaly detection, but mainly only focus on finding unusual data. To extend, we can expand the methods for better outlier removal.

7

CHAPTER

APPENDIX

CONTENTS

7.1 Method implementations	109
----------------------------	-----

7.1 Method implementations

This section presents a detailed and coherent implementation approach for the core algorithms/techniques used in Chapter 2.

7.1.1 Local descriptor - SIFT

In this work, we implemented Scale-invariant feature transform (SIFT) as local image descriptor. SIFT is one of the most widely implemented algorithm to detect and describe local image features. Despite SIFT is not the latest algorithm, it is robust to environment, widely recognised by computer vision researchers and easy to implement

due to its open source code. Also, since same image local descriptor is used across all method categories, the choice would not affect the final comparison result. A brief implementation walkthrough of SIFT keypoint detector and descriptor is given below.

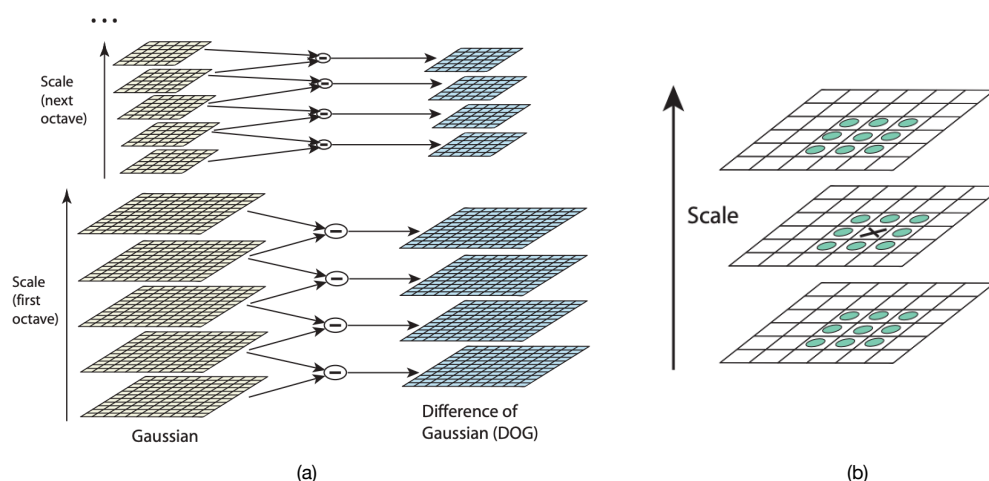


Figure 7.1: Figure 1 in the paper of [Lowe, 2004a]. Figure (a) shows how multi-scaled DoGs are calculated; figure (b) shows the approach for finding local maxima/minima of DoG.

Local interest points for SIFT description, or keypoints, are detected via scale-space extrema detection. This step involves three stages:

- Image convolution with Gaussian filter at multiple scales.

The convolution of the original image $I(x, y)$ with the Gaussian kernel $G(x, y, k\sigma)$ at scale $k\sigma$ is given by:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$$

, where $*$ indicates convolution operator

- Find difference of successive convolved images, as illustrated in Figure 7.1 (a).

Difference of Gaussians (DoG) $D(x, y, \sigma)$ at each scale is given by:

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$$

- Find local maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales, , as illustrated in Figure 7.1 (b).

This is given by the comparisons between each pixel in every DoG layers with its 8 neighbours on same scale and 9 neighbours in each of the neighbouring scales. The pixel is selected as local maxima/minima, i.e. a keypoint, if it greater or less than all of its neighbours.

After first step, there are likely to have lots of keypoints that have low repeatability or poorly located, i.e. they are no longer keypoints under slight environment variant such as illumination change or view point change. There are several aspects for improving and discarding keypoints:

- Sub-pixel determination of keypoints.

In the work by [Brown and Lowe, 2002], an approach improve the accuracy of keypoint localisation uses the Taylor expansion (up to the quadratic terms) of the scale-space function as below:

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

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

- Removing keypoints with low-contrast.

The contrast can be approximated by the value of the second-order Taylor expansion $D(\mathbf{x})$. According to the experiment of [Lowe, 2004a], the keypoints with value below 0.03 are discarded, assuming image pixel values are in the range of [0,1].

- Removing keypoints that have poorly determined locations but high edge responses.

This is done by calculating the principal curvature, the eigenvalues of a 2x2 Hessian matrix at the location and scale of each keypoint. Since only the ratio of principal curvatures is of interest, i.e. the ratio between the largest magnitude eigenvalue and the smaller one, this process is finally simplified to:

$$(D_{xx} + D_{yy})^2 / (D_{xx}D_{yy} - D_{xy}^2) > (r + 1)^2 / r,$$

where D is DoG function of keypoint (x, y) , r is a threshold of the ratio between largest and second largest eigenvalue. In the experiment of [Lowe, 2004a], $r = 10$.

To achieve scale-invariant, dominant orientations are calculated via functions below at given scale σ and Gaussian-blurred image $L(x, y, \sigma)$. At a location x, y :

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \arctan2(L(x, y + 1) - L(x, y - 1), L(x + 1, y) - L(x - 1, y)),$$

where, $m(x, y)$ is the gradient magnitude, $\theta(x, y)$ is the orientation. In the paper, an orientation histogram is formed with 36 bins with 10 degree coverage each bin, weighted by the gradient magnitude. The histogram peak is selected as the dominant orientation.

The next stage is local description for all keypoints at particular scales and assigned orientations. The descriptor is aimed to be highly distinctive and nuisance invariant. Similar to the calculation of dominant orientations, each SIFT descriptor consists of a 4x4 histogram of orientation with 8 bins each, as illustrated in Figure 7.2. This makes the feature vector 128-dimensions in total, a unified representation of local image keypoints. Image patches similarity can be simply measured from Euclidean distance between feature vectors, or via approximate nearest neighbour algorithms for large-scaled problem.

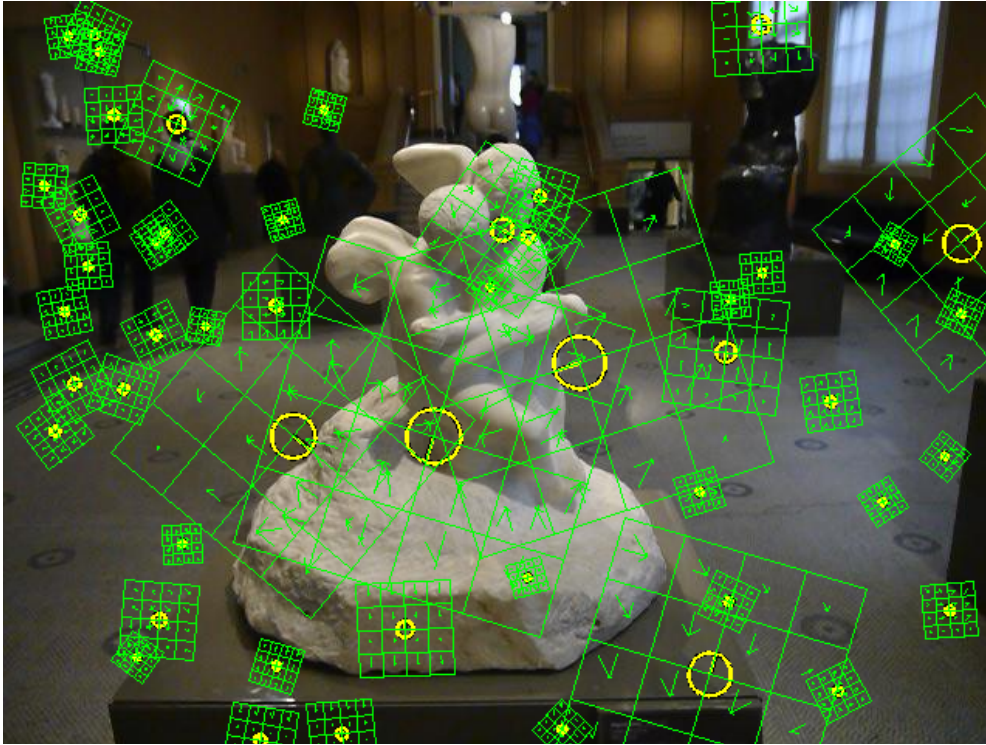


Figure 7.2: Example of SIFT descriptor on a sample image of our proposed dataset.

7.1.2 Global descriptor - Bag-of-Words

Unlike image local descriptor, global descriptor aims to encode image as a single representation, hence enables the similarity measurement between images. We adopt the bag-of-words approach for comparison in this work, which the representation is encoded based on the set of local image descriptors.

Bag-of-Words model (BoW) is commonly applied for document classification or other natural language processing problems. It discards the grammar, words order and other context information, but takes the occurrence of each word as feature to train a classifier. This process results a histogram of a fixed number of texts, so that the

similarity between documents can be measured by the distance between histograms.

In computer vision, the text words are replaced by the local descriptor. However, since local descriptors are scattered in the feature space, we have to take an extra stage to generate a dictionary, or a ‘codebook’, to map the local descriptor to ‘codewords’. This step is usually achieved by k-means clustering over all descriptors. A brief workflow of BoW approach is illustrated in the Figure 7.3.

At the end, each descriptor is mapped to a codewords, so that a histogram can be generated from the set of local descriptors, which represents the whole image.

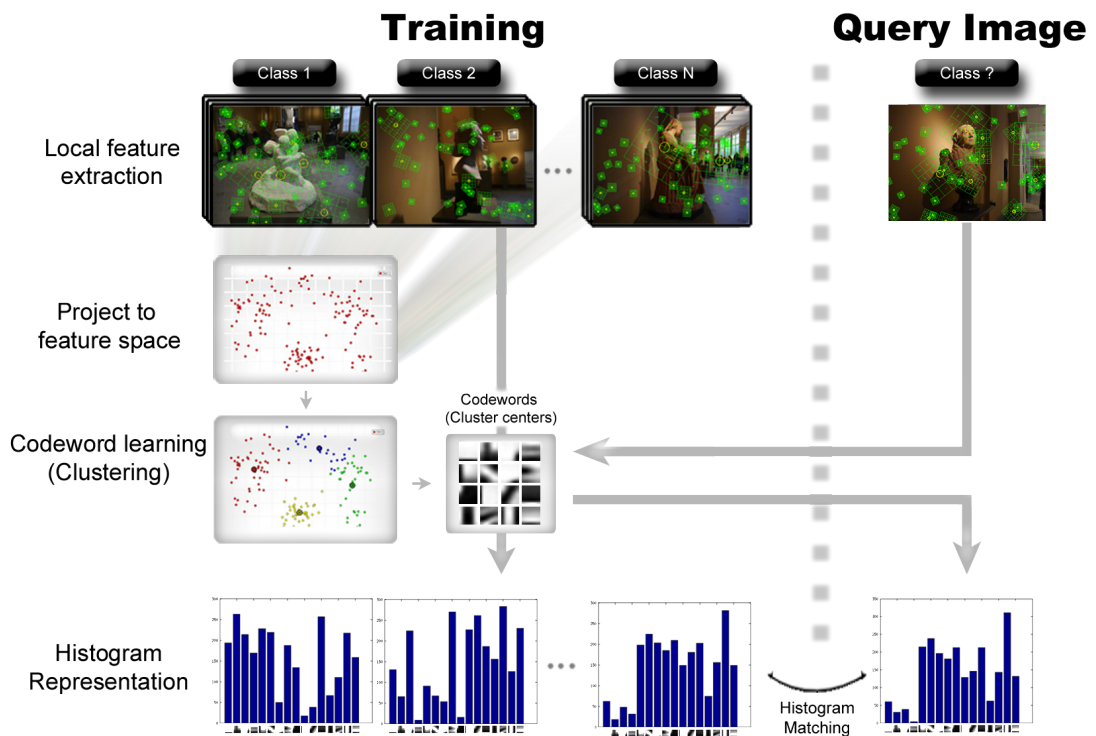


Figure 7.3: A brief workflow of Bag-of-Words approach.

BoW has its limitations from discarding the spatial relationships among the image patches. This is improved by the work of Spatial Pyramid Matching (SPM) by [Lazebnik et al., 2006]. The key of this approach is to coarsely encode the spatial information

in the image encoding, so that the spatial relationship is not completely lost during the BoW process.

The implementation of SPM is quite simple: applying BoW encoding on different regions (with different scales) and then concatenate all histograms, a toy example from its original paper is illustrated in the Figure 7.4.

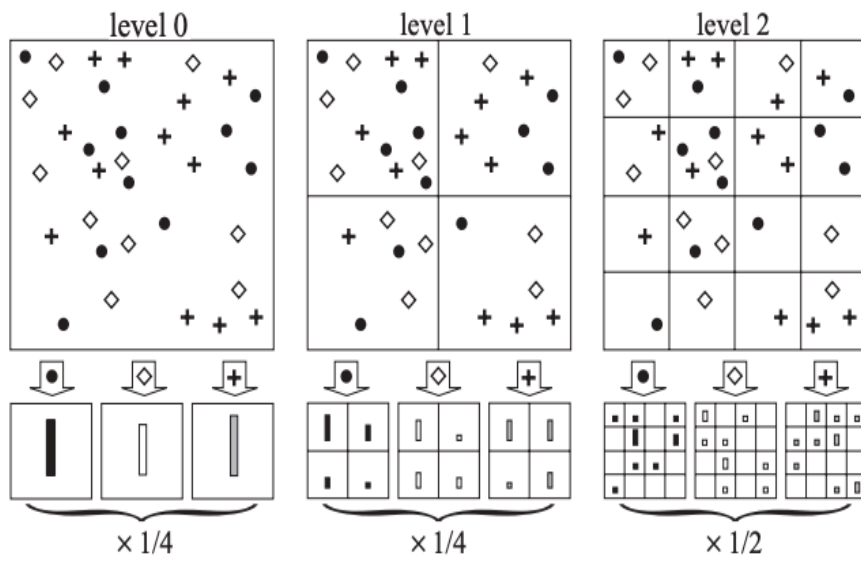


Figure 7.4: A toy example from Figure 1 of the work [Lazebnik et al., 2006]. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, the image is subdivided at three different levels of resolution. At each level of resolution, a histogram is formed and weighted. The final representation takes the concatenation of all histograms.

We also implemented a better encoding scheme, called Sparse Coding, also known as ScSPM when combining with SPM, this work is proposed by [Yang et al., 2009]. This work replaces the vector quantisation step in the SPM from k-means clustering to sparse coding. Instead of ‘hard’ assigning of local descriptor to a single ‘codeword’, sparse coding learns a set of basis so that each local descriptor can be ‘soft’ assigned to a combination of basis with various weight. This greatly reduce the information loss

from vector quantisation hence improve the representation of images. The details of training process can be referred to the section 3.1 in the paper [Yang et al., 2009].

7.1.3 Approximate nearest neighbour

Approximate nearest neighbour (ANN) algorithm is implemented to accelerate the large-scale pairwise matching between local/global image descriptors, it can accelerate nearest neighbour matching by hundreds times over exact search, as shown in Figure 7.5. In this work, we implemented the Fast Library for Approximate Nearest Neighbours (FLANN) algorithm by [Muja and Lowe, 2014]. FLANN is a recent popular and open-sourced library for very efficient high dimensional nearest neighbour matching. It is based on the randomised k-d forest with a novel node partitioning algorithm, called priority search k-means forest.

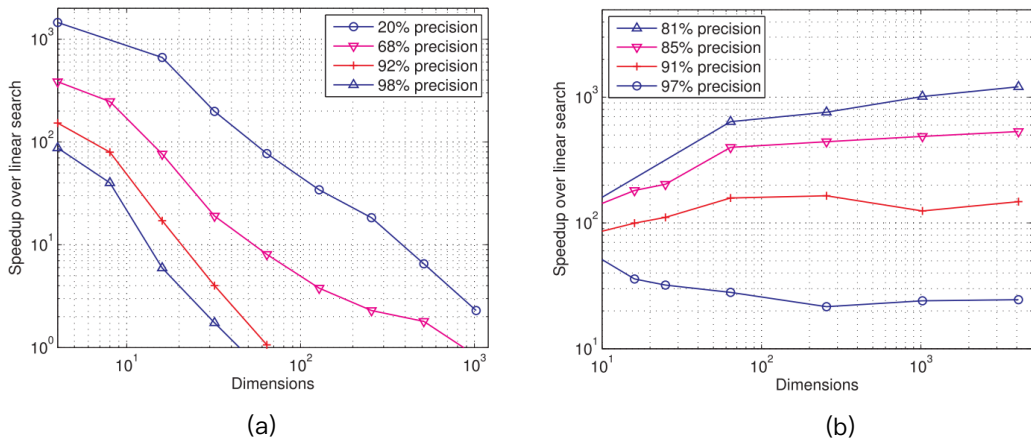


Figure 7.5: An experiment conducted in the work [Muja and Lowe, 2014]. Search efficiency for data of varying dimensionality, with data set of size 100K. (a) uses data with no correlations (random vectors), (b) uses real-world image descriptors.

To present the workflow of the construction of a randomised k-d forest, we start with a single randomised k-d tree. Basically, the dataset is recursive split into two (or

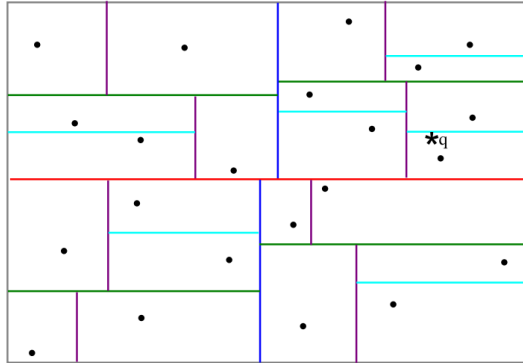


Figure 7.6: An example of kd-tree from work [Muja and Lowe, 2014]. The data is split until each leaf node has one data point.

multiple) subsets, following a splitting rule, each subset will be further split until they cannot be further split.

Before the construction, we need to set a stopping criteria to stop further partition of nodes under a pre-defined condition, it can be a maximum number of layers or data in the leaf nodes. Then we also set a maximum number of iterations we will perform the random subspace method on each split node.

We start establishing the tree by splitting from the root node with full dataset. For each node, a hyperplane need to be chosen to split the dataset in a node into two subsets for child nodes. There are many ways to find out the hyperplane. In the work of [Muja and Lowe, 2014], they first calculate the top 5 dimensions with the highest data variance and choose one of them randomly. After the hyperplane is learnt, the data can be split into two subset by simply compare each of data with the plane. A simple example with 2D data is shown in Figure 7.6. For a query point, it only compares with the hyperplanes to reach its approximate nearest neighbour.

Since there is likely to exist a nearest neighbour that is across a decision boundary

from the query point. K-d tree is trained multiple times to form a randomised k-d forest, each tree is built with different random seeds to improve the matching precision.

7.1.4 RANSAC

Random sample consensus method (RANSAC) is used to estimate the parameters of a model that fit a set of inliers that mixed with a (often much larger) set of outliers. The algorithm is implemented according to [Bolles and Fischler, 1981] and briefly summarised as below:

- Randomly select a set of hypothetical inliers, usually is a minimal subset of the data that can fit the model.
- Estimate model parameters from hypothetical inliers.
- Fit all data through the model parameter, count the inlier data points that according to a model-specific loss function, e.g. a Euclidean distance threshold between an estimated data point and the corresponding true data point.
- Repeat the procedure until certain condition is met, e.g. reaching a maximum number of iteration or finding a model parameter that fits enough data points.
- The final model parameters might re-estimated from the full inlier set instead of the hypothetical inliers.

In this work, RANSAC is applied to estimate the geometric relationship between two sets of SIFT descriptors for filtering the outliers that do not pass the geometric validation, and also to find the relative camera pose between two images for 3D reconstruction.

7.1.5 Camera calibration



Figure 7.7: A camera calibration board.

Since our work involves 2D-to-3D projection and 3D reconstruction, camera intrinsic parameters are required to map pixel coordinates to world coordinates:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix},$$

where A is the camera intrinsic matrix, $[u \ v \ 1]^T$ is the 2D points in pixel coordinates, $[x_w \ y_w \ z_w \ 1]^T$ is the 3D points in world coordinates, R, T are the extrinsic parameters

that denote the transformation between 3D world coordinates and 3D camera coordinates, i.e. the camera pose.

A camera intrinsic matrix A is:

$$A = \begin{bmatrix} f_u & \gamma & u_c & 0 \\ 0 & f_v & v_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

where f_u, f_v represent the focal length on both axes, u_c, v_c represent the principal point, γ is the skew coefficient between axes.

To compute camera intrinsic parameters, we first acquire a calibration board with checkerboard pattern, as illustrated in Figure 7.7. The physical size of checkerboard is known in world units, e.g. millimetres. Then for an image of checkerboard, we can extract the corner points by corner detector (such as Harris corner detector [Harris and Stephens, 1988]) and find their correspondences by RANSAC or other method.

The algorithm for computing camera intrinsic parameters is done as below:

- Start with an initial intrinsic parameters.
- Take an image of the calibration board and estimate the reprojection error by solving the Perspective-n-Point problem. The implementation of this method is introduced in the next subsection.
- Run a global optimisation algorithm, e.g. Levenberg-Marquardt, with respect to the camera intrinsic parameters to minimise the reprojection error.

After taking ten or twenty images of checkerboard with various viewpoints, the error should converge to a small value and return the final values of camera intrinsic parameters.

7.1.6 Perspective-n-Point problem

Perspective-n-Point problem (PnP) is to find the camera pose (position and orientation) given the camera intrinsic parameters and a set of n correspondences between 3D points and their 2D projections. We implemented a fast method called ePnP by [Lepetit et al., 2009] to validate the 2D-to-3D geometry between an object image and its 3D object model, an example of usage is given in the Figure 7.8.

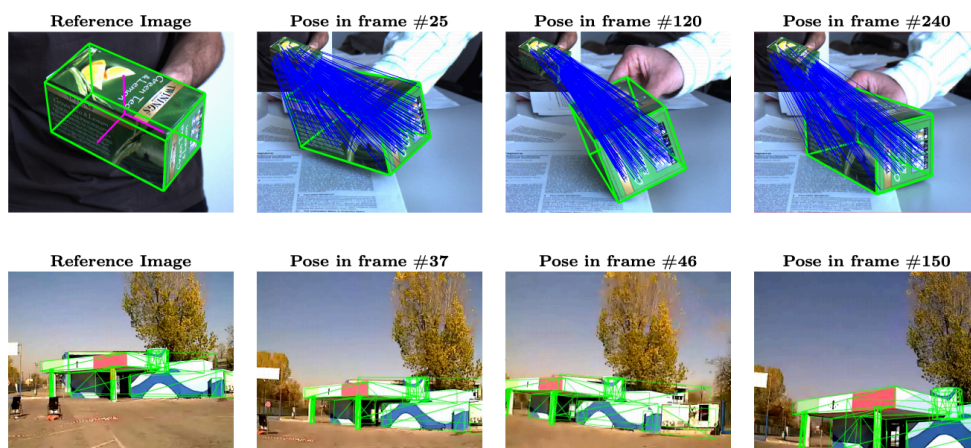


Figure 7.8: From [Lepetit et al., 2009]. Examples of reprojection of the models on real images.

Given a 3D object model, i.e. a point cloud that consists of n 3D points in the world coordinate system, be:

$$[p_1, p_2, \dots, p_n]$$

And let 4 control points be:

$$[c_1, c_2, c_3, c_4]$$

Firstly we represent all 3D points by control points, such that:

$$p_i = \sum_{j=1}^4 \alpha_{ij} c_j, \text{ with } \sum_{j=1}^4 \alpha_{ij} = 1,$$

where the α_{ij} are homogeneous barycentric coordinates.

Next, let A be the camera intrinsic matrix and $\{(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)\}$ be the 2D projections of the $\{p_1, p_2, \dots, p_n\}$. We have:

$$Ap_i^c = w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}, \text{ where } A = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix}, w_i \text{ are scalar projective parameters}$$

In intrinsic matrix A , f_u, f_v are focal length coefficients and the u_c, v_c are principal points. For 3D coordinates, let control point $c_i = [x_j^c, y_j^c, z_j^c]^T$, the last equation becomes:

$$w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^4 a_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix}$$

From the last row of this equation, we can see that $w_i = \sum_{j=1}^4 a_{ij} z_j^c$. By substituting it to the previous equation, we have:

$$\begin{aligned} \sum_{j=1}^4 a_{ij} f_u x_j^c + a_{ij} (u_c - u_i) z_j^c &= 0 \\ \sum_{j=1}^4 a_{ij} f_v y_j^c + a_{ij} (v_c - v_i) z_j^c &= 0 \end{aligned}$$

By concatenating two equations for all reference points, we can generate a linear system:

$$M\mathbf{x} = 0,$$

where $\mathbf{x} = [c_1^T, c_2^T, c_3^T, c_4^T]^T$, M is a $2n \times 12$ matrix, generated by the coefficients from previous two equations.

The solution is expressed as a linear combination of the null eigenvectors of $M^T M$, a constant matrix of (12×12) size. According to the experiments in [Lepetit et al.,

2009], this step consumes most of the computation power. The solution is expressed as below:

$$\mathbf{x} = \sum_{i=1}^N \beta_i v_i,$$

where the set v_i are the columns of the right-singular vectors of M corresponding to the N null singular values of M .

The dimension N of the null-space depends on the camera model, i.e. how many ambiguities exist in the solution. Given more than 6 correspondent points and a perspective camera model, $N = 1$ because of the scale ambiguity; for an affine camera model, $N = 4$ because of the unknown depths of the four control points. Since a perspective camera with a large focal length can be approximated to an affine camera, the value of N is not certain. In the paper of [Lepetit et al., 2009], the authors compute all values of N and keep the one with smallest reprojection error.

In our implementation, we only consider the case that camera model is perspective and there are more than 6 correspondent points, i.e. $N = 1$. This is because our camera positions in our problem setting are generally near to the objects, and the camera has relatively small focal length. The camera pose estimation is considered failure if reprojection error is higher than a certain threshold.

7.1.7 Set similarity - Kernel principal angle

We adopt kernel principal angle method to measure the similarity between two subspaces (or manifold), each subspace is formed with a set of local or global image descriptors, the principal angles are invariant to the column ordering of the two sets. This method is implemented follow the work of [Wolf and Shashua, 2003].

Let two descriptor sets $A = [\phi(\mathbf{a}_1), \dots, \phi(\mathbf{a}_k)]$ and $B = [\phi(\mathbf{b}_1), \dots, \phi(\mathbf{b}_k)]$ represent two linear subspaces U_A, U_B in the feature space where $\phi(\cdot)$ is mapping from input space R^n onto a feature space F with a kernel function $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$. The principal angles $0 < \theta_1 < \dots < \theta_k \leq (\pi/2)$ between the two subspaces are defined as:

$$\cos(\theta_k) = \max_{\mathbf{u} \in U_A} \max_{\mathbf{v} \in U_B} \mathbf{u}^T \mathbf{v}$$

subject to:

$$\mathbf{u}^T \mathbf{u} = \mathbf{v}^T \mathbf{v} = 1, \mathbf{u}^T \mathbf{u}_i = 0, \mathbf{v}^T \mathbf{v}_i = 0, i = 1, \dots, k - 1$$

$\cos(\theta_i)$ are the canonical correlations of the matrix pair (A, B) . The vectors $\mathbf{u}_i, \mathbf{v}_i$ are called variates and the corresponding vectors $\mathbf{x}_i, \mathbf{y}_i, \mathbf{u}_i = A\mathbf{x}_i$ and $\mathbf{v}_i = B\mathbf{y}_i$ are called the canonical vectors.

To find out $\cos(\theta_i)$, we follow the eigen decomposition approach. The eigen decomposition U_A, D_A, U_B, D_B is calculated using SVD formulation $A^T A U_A = U_A D_A$ and $B^T B U_B = U_B D_B$. Then let $M = A^T B$, the cosine of the principal angles are the singular values of the matrix $D_A^{-1/2} U_A^T M U_B D_B^{-1/2}$.

We adopt the positive definite kernel according to the recommendation in [Wolf and Shashua, 2003], as the final value to represent the set similarity:

$$f(A, B) = \prod_{i=1}^k \cos(\theta_i)^2 = \det(Q_A^T Q_B)^2$$

7.1.8 Tracking - Optical flow

We adopt sparse optical flow as our tracking method to form local descriptor trajectories along the video stream. Specifically, we adopt Lucas-Kanade method (KLT) by [Lucas et al., 1981] with bidirectional validation by [Kalal et al., 2010]. This methods work

fast and has good performance for small movement, but likely to fail under large displacement or long time tracking. As our target video clips are relatively short but high frame rate, this method suits our requirement. The local point tracking is illustrated in the Figure 7.9.

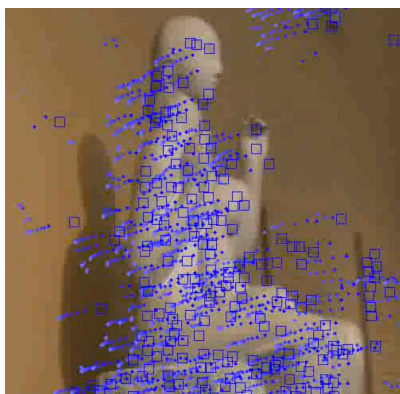


Figure 7.9: An example of local points tracking. The square boxes are the local points in current frame, the dots are the local points in previous frames.

Given a 2D location in a video p at frame t with image intensity $I(p, t)$, we aim to find the movement $(\Delta x, \Delta y)$ of each position such that:

$$I(p, t) = I(p - (\Delta x, \Delta y), t - 1)$$

By assuming all the neighbouring pixels of a given position will have similar motion, the problem becomes solving nine equation in total with two unknown variables:

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\ &\dots \\ I_x(q_9)V_x + I_y(q_9)V_y &= -I_t(q_9), \end{aligned}$$

where q are all known neighbouring pixels given a local image location p , (V_x, V_y) are local image velocities that are equivalent to $(\Delta x, \Delta y)$ in this case. $I_x(q_i), I_y(q_i), I_t(q_i)$ are the partial derivatives of the image I with respect to position x, y and time t , i.e. the I_x, I_y can be calculated by subtracting $I(p) - I(q_i)$, I_t is the intensity values difference of local images between frame t and $t - 1$.

Hence the movement $\Delta x, \Delta y$ is computed by solving the 2×2 system:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_y(q_i)I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix}$$

7.1.9 3D Reconstruction - Structure from motion (Off-the-shelf)

Given local correspondences between two images and a camera intrinsic parameter, one can estimate the relative camera pose hence project the 2D local points on image (camera coordinates) into 3D world coordinates. The implementation in this work is based on an off-the-shelf GUI application 'VisualSFM' by [Wu, 2011].

To reconstruct a sparse 3D point cloud from a set of object videos, the workflow is shown below:

- Do camera calibration to find the camera intrinsic parameters.
- Extract SIFT features from every image.
- Perform approximate nearest neighbour algorithm (FLANN) to find correspondence between images. Note that to reduce the complexity of pairwise matching, only the images within 5 seconds are matched to each other. To overcome loop closure, the features of a set of randomly picked images are also pairwise matched.

- Feed the feature information of images and their matches into VisualSfM. Each match consists of two feature indices; each image is attached with their feature locations.
- Run the program and export camera 6-dof poses and reconstructed 3D points.
- The 3D points are correspondent to the 2D features so that they can be used for further 2D-to-3D or 3D-to-3D matching.

Dense point clouds are extracted by replacing sparse SIFT descriptors with dense optical flow trajectories, the correspondences are filtered with RANSAC. The result is shown in Figure 7.10.

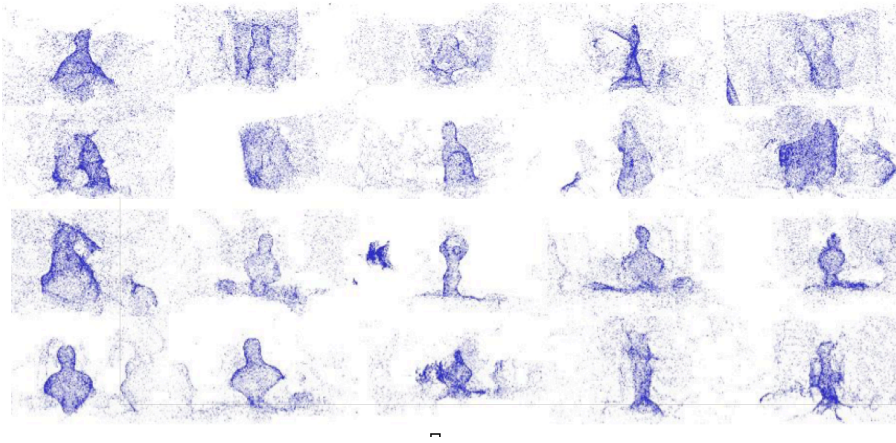


Figure 7.10: Reconstructed 3D point cloud from our proposed dataset.

7.1.10 Nearest Neighbour-based classifier

Nearest Neighbour-based classifier (or Naive-Bayes Nearest-Neighbour, NBNN) is a simple, efficient non-parametric classifier proposed by [Boiman et al., 2008]. It does not need the learning phase and can handle large number of classes. The authors prove

that under the Naive-Bayes assumption, the optimal distance to use in image classification is the Kullback-Leibler (KL) “Image-to-Class” distance, and not the commonly used “Image-to-Image” distribution distances.

Given a query descriptor Q , and a target class C , the maximum-a-posteriori (MAP) classifier minimises the average classification error:

$$\hat{C} = \arg \max_C p(C|Q),$$

when the class prior $p(C)$ is uniform, the MAP classifier reduces to the Maximum Likelihood (ML) classifier:

$$\hat{C} = \arg \max_C p(Q|C),$$

since the descriptors are i.i.d., follow the Naive-Bayer assumption:

$$p(Q|C) = p(d_1, \dots, d_n|C) = \prod_{i=1}^n p(d_i|C),$$

take the log probability:

$$\hat{C} = \arg \max_C p(Q|C) = \arg \max_C \frac{1}{n} \sum_{i=1}^n \log p(d_i|C),$$

then reform to:

$$\hat{C} = \arg \max_C \left(\sum_{d \in D} p(d|Q) \log \frac{p(d|C)}{p(d|Q)} \right) = \arg \min_C (KL(p(d|Q)||p(d|C)))$$

where $KL(p(d|Q)||p(d|C))$ is the KL-distance (divergence) between two probability distributions. This proves that the optimal MAP classifier minimises a “Query-to-Class” KL-distance between the descriptor distributions of the query Q and the class C .

The probability density of descriptor d in a class C is required, i.e. $p(d|C)$ to compute the classification error, however, the number of descriptors is too large. Parzen density estimation is adopted to approximate the continuous descriptor probability density $p(d|C)$, follow by [Duda et al., 2012]:

$$\hat{p}(d|C) = \frac{1}{L} \sum_{j=1}^L K(d - d_j^C),$$

where K is the Parzen kernel function, L is the number of descriptors. The approximation \hat{p} converges to the p as L approaches infinity.

From the long-tail characteristic of descriptor distributions, as descriptors are mostly isolated in the feature space, we can further assume that after certain r nearest neighbour, the K is negligible, i.e. :

$$p_{NN}(d|C) = \frac{1}{L} \sum_{j=1}^r K(d - d_{NN_j}^C)$$

From the work of [Boiman et al., 2008], this approximation is accurate enough even for $r = 1$. Therefore the implementation of algorithm can be simplified as below:

- Given a set of query descriptors D .
- Compute the nearest neighbours of all descriptors d in D in all C : $NN_C(d)$.
- $\hat{C} = \arg \min_C \sum_{i=1}^n ||d_i - NN_C(d_i)||^2$

In this work, we also implemented an extension of NBNN, called Local-NBNN, by [McCann and Lowe, 2012]. It has better scalability to the number of class, and achieves 100 times speed-up over the original NBNN on the Caltech-256 dataset.

The implementation is given below:

- Given a set of query descriptors D .
- Compute $k + 1$ nearest neighbours p of all descriptors d in D in all C : $NN_C(d, k + 1)$.
- Find the background distance: $dist_B = ||d_i - p_{k+1}||^2$.
- For all categories C found in the k nearest neighbours:

$$dist_C = \min_{\{p_j | Class(p_j)=C\}} ||d_i - p_j||^2$$

$$totals[C] \leftarrow total[C] + dist_C - dist_B$$

- $\hat{C} = \arg \min_C \text{totals}[C]$

BIBLIOGRAPHY

- [Aggarwal, 2015] Aggarwal, C. C. (2015). Outlier analysis. In *Data mining*, pages 237–263. Springer.
- [Alcantarilla et al., 2012] Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). Kaze features. In *Computer Vision–ECCV 2012*, pages 214–227. Springer.
- [Alexe et al., 2010] Alexe, B., Deselaers, T., and Ferrari, V. (2010). What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE.
- [Angiulli and Fassetti, 2007] Angiulli, F. and Fassetti, F. (2007). Very efficient mining of distance-based outliers. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 791–800. ACM.
- [Bao et al., 2012] Bao, S. Y., Bagra, M., Chao, Y.-W., and Savarese, S. (2012). Semantic structure from motion with points, regions, and objects. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2703–2710. IEEE.
- [Bay et al., 2000] Bay, S. D., Kibler, D. F., Pazzani, M. J., and Smyth, P. (2000). The uci kdd archive of large data sets for data mining research and experimentation. *SIGKDD explorations*, 2(2):81–85.
- [Behmo et al., 2010] Behmo, R., Marcombes, P., Dalalyan, A., and Prinet, V. (2010). Towards optimal naive bayes nearest neighbor. In *European Conference on Computer Vision (ECCV)*, pages 171–184. Springer.

- [Beis and Lowe, 1997] Beis, J. S. and Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1000–1006. IEEE.
- [Boiman et al., 2008] Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE.
- [Bolles and Fischler, 1981] Bolles, R. C. and Fischler, M. A. (1981). A ransac-based approach to model fitting and its application to finding cylinders in range data. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 1981, pages 637–643.
- [Bourdev et al., 2011] Bourdev, L., Maji, S., and Malik, J. (2011). Describing people: A poselet-based approach to attribute classification. In *2011 International Conference on Computer Vision*, pages 1543–1550. IEEE.
- [Bouwmans and Zahzah, 2014] Bouwmans, T. and Zahzah, E. H. (2014). Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34.
- [Brachmann et al., 2014] Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., and Rother, C. (2014). Learning 6d object pose estimation using 3d object coordinates. In *European Conference on Computer Vision*, pages 536–551. Springer.
- [Branch et al., 2013] Branch, J. W., Giannella, C., Szymanski, B., Wolff, R., and Kargupta, H. (2013). In-network outlier detection in wireless sensor networks. *Knowledge and information systems*, 34(1):23–54.
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- [Brown and Lowe, 2002] Brown, M. and Lowe, D. G. (2002). Invariant features from interest point groups. In *BMVC*, volume 4.

- [Bucak et al., 2014] Bucak, S. S., Jin, R., and Jain, A. K. (2014). Multiple kernel learning for visual object recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1354–1369.
- [Chernov and Lesort, 2004] Chernov, N. and Lesort, C. (2004). Statistical efficiency of curve fitting algorithms. *Computational statistics & data analysis*, 47(4):713–728.
- [Cho and Lee, 2012] Cho, M. and Lee, K. M. (2012). Progressive graph matching: Making a move of graphs via probabilistic voting. In *Computer Vision and Pattern Recognition (CVPR)*, pages 398–405. IEEE.
- [Chum and Matas, 2002] Chum, O. and Matas, J. (2002). Randomized ransac with td, d test. In *Proc. British Machine Vision Conference*, volume 2, pages 448–457.
- [Chum et al., 2003] Chum, O., Matas, J., and Kittler, J. (2003). Locally optimized ransac. In *Pattern Recognition*, pages 236–243. Springer.
- [Collet et al., 2011] Collet, A., Martinez, M., and Srinivasa, S. S. (2011). The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research (IJRR)*, 30(10):1284–1306.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- [De la Torre et al., 2008] De la Torre, F., Hodgins, J., Bargteil, A., Martin, X., Macey, J., Collado, A., and Beltran, P. (2008). Guide to the carnegie mellon university multi-modal activity (cmu-mmact) database.
- [Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE.

- [Deng et al., 2018] Deng, J., Guo, J., and Zafeiriou, S. (2018). Arcface: Additive angular margin loss for deep face recognition. *arXiv: Computer Vision and Pattern Recognition*.
- [Désir et al., 2013] Désir, C., Bernard, S., Petitjean, C., and Heutte, L. (2013). One class random forests. *Pattern Recognition*, 46(12):3490–3506.
- [Drost et al., 2010] Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010). Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, volume 1, page 5.
- [Duchenne et al., 2011] Duchenne, O., Joulin, A., and Ponce, J. (2011). A graph-matching kernel for object categorization. In *International Conference on Computer Vision (ICCV)*, pages 1792–1799. IEEE.
- [Duda et al., 2012] Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.
- [Dutta et al., 2007] Dutta, H., Giannella, C., Borne, K., and Kargupta, H. (2007). Distributed top-k outlier detection from astronomy catalogs using the demac system. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 473–478. SIAM.
- [Eberly, 1998] Eberly, D. (1998). Distance from a point to an ellipse, an ellipsoid, or a hyperellipsoid. Technical report, Technical report, Geometric Tools, LLC, 2011. 5.1.3.
- [Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338.
- [Fei-Fei and Perona, 2005] Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531. IEEE.

- [Fischler and Bolles, 1981a] Fischler, M. A. and Bolles, R. C. (1981a). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- [Fischler and Bolles, 1981b] Fischler, M. A. and Bolles, R. C. (1981b). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- [Fitzgibbon et al., 1999] Fitzgibbon, A., Pilu, M., and Fisher, R. B. (1999). Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):476–480.
- [Fitzgibbon, 2003] Fitzgibbon, A. W. (2003). Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13):1145–1153.
- [Forsyth and Ponce, 2002] Forsyth, D. A. and Ponce, J. (2002). *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference.
- [Funkhouser et al., 2003] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D. (2003). A search engine for 3d models. *ACM Transactions on Graphics (TOG)*, 22(1):83–105.
- [Ghoting et al., 2006] Ghoting, A., Parthasarathy, S., and Otey, M. E. (2006). Fast mining of distance-based outliers in high-dimensional datasets. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 609–613. SIAM.
- [Goodman et al., 2000] Goodman, J. E., O’Rourke, J., and Rosen, K. H. (2000). *Handbook of discrete and computational geometry*. cRc Press LLC.
- [Gordon and Lowe, 2006a] Gordon, I. and Lowe, D. G. (2006a). What and where: 3d object recognition with accurate pose. In *Toward category-level object recognition*, pages 67–82. Springer.

- [Gordon and Lowe, 2006b] Gordon, I. and Lowe, D. G. (2006b). What and where: 3d object recognition with accurate pose. In *Toward category-level object recognition*, pages 67–82. Springer.
- [Hao et al., 2013a] Hao, Q., Cai, R., Li, Z., Zhang, L., Pang, Y., Wu, F., and Rui, Y. (2013a). Efficient 2d-to-3d correspondence filtering for scalable 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 899–906.
- [Hao et al., 2013b] Hao, Q., Cai, R., Li, Z., Zhang, L., Pang, Y., Wu, F., and Rui, Y. (2013b). Efficient 2d-to-3d correspondence filtering for scalable 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 899–906. IEEE.
- [Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey Vision*, volume 15, page 50. Manchester, UK.
- [Hauskrecht et al., 2013] Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G. F., and Clermont, G. (2013). Outlier detection for patient monitoring and alerting. *Journal of Biomedical Informatics*, 46(1):47–55.
- [Hawkins, 1980] Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.
- [Hinterstoisser et al., 2012a] Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., and Lepetit, V. (2012a). Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888.
- [Hinterstoisser et al., 2012b] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., and Navab, N. (2012b). Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer.

- [Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network. *ArXiv e-prints*.
- [Hirschmuller, 2008] Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):328–341.
- [Idris et al., 2015] Idris, I., Selamat, A., Nguyen, N. T., Omatu, S., Krejcar, O., Kuca, K., and Penhaker, M. (2015). A combined negative selection algorithm–particle swarm optimization for an email spam detection system. *Engineering Applications of Artificial Intelligence*, 39:33–44.
- [Irschara et al., 2009] Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2599–2606. IEEE.
- [Ismo et al., 2004] Ismo, K. et al. (2004). Outlier detection using k-nearest neighbour graph. In *null*, pages 430–433. IEEE.
- [Johnson and Hebert, 1999] Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449.
- [Kalal et al., 2010] Kalal, Z., Mikolajczyk, K., and Matas, J. (2010). Forward-backward error: Automatic detection of tracking failures. In *International Conference on Pattern Recognition (ICPR)*, pages 2756–2759. IEEE.
- [Kanatani and Rangarajan, 2010] Kanatani, K. and Rangarajan, P. (2010). Hyperaccurate ellipse fitting without iterations. In *VISAPP (2)*, pages 5–12.
- [Kehl et al., 2015] Kehl, W., Tombari, F., Navab, N., Ilic, S., and Lepetit, V. (2015). Hashmod: A Hashing Method for Scalable 3D Object Detection. In *Proceedings of the British Machine Vision Conference*.

- [Knopp et al., 2010] Knopp, J., Prasad, M., Willems, G., Timofte, R., and Van Gool, L. (2010). Hough transform and 3d surf for robust three dimensional classification. In *European Conference on Computer Vision (ECCV)*, pages 589–602. Springer.
- [Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178. IEEE.
- [Lepetit et al., 2009] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision (IJCV)*, 81(2):155–166.
- [Li and DiCarlo, 2010] Li, N. and DiCarlo, J. J. (2010). Unsupervised natural visual experience rapidly reshapes size invariant object representation in inferior temporal cortex. *Neuron*, 67(6):1062–1075.
- [Lin et al., 2012] Lin, L., Luo, P., Chen, X., and Zeng, K. (2012). Representing and recognizing objects with massive local image patches. *Pattern Recognition*, 45(1):231–240.
- [Lindeberg, 2013] Lindeberg, T. (2013). Invariance of visual operations at the level of receptive fields. *BMC Neuroscience*, 14(Suppl 1):P242.
- [Liu et al., 2011] Liu, X., Lin, L., Yan, S., Jin, H., and Tao, W. (2011). Integrating spatio-temporal context with multiview representation for object recognition in visual surveillance. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(4):393–407.
- [Liu et al., 2014] Liu, Y., Jang, Y., Woo, W., and Kim, T.-K. (2014). Video-based object recognition using novel set-of-sets representations. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 533–540. IEEE.
- [Lowe, 2004a] Lowe, D. G. (2004a). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110.

- [Lowe, 2004b] Lowe, D. G. (2004b). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.
- [Lucas et al., 1981] Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 81, pages 674–679.
- [MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA.
- [McCann and Lowe, 2012] McCann, S. and Lowe, D. G. (2012). Local naive bayes nearest neighbor for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3650–3656. IEEE.
- [Mei et al., 2011] Mei, L., Liu, J., Hero, A., and Savarese, S. (2011). Robust object pose estimation via statistical manifold modeling. In *International Conference on Computer Vision (ICCV)*.
- [Muja and Lowe, 2014] Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36.
- [Ng and Winkler, 2014a] Ng, H. and Winkler, S. (2014a). A data-driven approach to cleaning large face datasets. pages 343–347.
- [Ng and Winkler, 2014b] Ng, H.-W. and Winkler, S. (2014b). A data-driven approach to cleaning large face datasets. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 343–347. IEEE.
- [Nistér, 2005a] Nistér, D. (2005a). Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329.

- [Nistér, 2005b] Nistér, D. (2005b). Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329.
- [Noceti et al., 2009] Noceti, N., Delponte, E., and Odone, F. (2009). Spatio-temporal constraints for on-line 3d object recognition in videos. *Computer Vision and Image Understanding (CVIU)*, 113(12):1198–1209.
- [Olaru and Wehenkel, 2003] Olaru, C. and Wehenkel, L. (2003). A complete fuzzy decision tree technique. *Fuzzy sets and systems*, 138(2):221–254.
- [Parashar et al., 2017] Parashar, A., Rhu, M., Mukkara, A., Puglielli, A., Venkatesan, R., Khailany, B., Emer, J. S., Keckler, S. W., and Dally, W. J. (2017). Scnn: An accelerator for compressed-sparse convolutional neural networks. *international symposium on computer architecture*, 45(2):27–40.
- [Peterson and Rhodes, 2003] Peterson, M. A. and Rhodes, G. (2003). *Perception of faces, objects, and scenes*. Oxford University Press New York.
- [Prasad et al., 2013] Prasad, D. K., Leung, M. K., and Quek, C. (2013). Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46(5):1449–1465.
- [Raguram et al., 2008] Raguram, R., Frahm, J.-M., and Pollefeys, M. (2008). A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision*, pages 500–513. Springer.
- [Rätsch et al., 2002] Rätsch, G., Mika, S., Scholkopf, B., and Müller, K.-R. (2002). Constructing boosting algorithms from svms: an application to one-class classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9):1184–1199.
- [Ren et al., 2017] Ren, S., He, K., Girshick, R. B., and Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

- [Ren and Gu, 2010] Ren, X. and Gu, C. (2010). Figure-ground segmentation improves handled object recognition in egocentric video. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3137–3144. IEEE.
- [Ren and Philipose, 2009] Ren, X. and Philipose, M. (2009). Egocentric recognition of handled objects: Benchmark and analysis. In *Computer Vision and Pattern Recognition Workshops (CVPR-WS)*, pages 1–8. IEEE.
- [Rios-Cabrera and Tuytelaars, 2013] Rios-Cabrera, R. and Tuytelaars, T. (2013). Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2048–2055.
- [Rothganger et al., 2006] Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2006). 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259.
- [Sattler et al., 2012] Sattler, T., Leibe, B., and Kobbelt, L. (2012). Improving image-based localization by active correspondence search. In *European Conference on Computer Vision (ECCV)*, pages 752–765. Springer.
- [Savarese and Fei-Fei, 2007] Savarese, S. and Fei-Fei, L. (2007). 3d generic object categorization, localization and pose estimation. In *International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE.
- [Schmidhuber et al., 1997] Schmidhuber, J., Zhao, J., and Wiering, M. (1997). Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement. *Machine Learning*, 28(1):105–130.

- [Schölkopf et al., 1999] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (1999). Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588.
- [Sivic and Zisserman, 2009] Sivic, J. and Zisserman, A. (2009). Efficient visual search of videos cast as text retrieval. *Pattern Analysis and Machine Intelligence (PAMI)*, 31(4):591–606.
- [Szeliski, 2010] Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
- [Tan, 2018] Tan, P.-N. (2018). *Introduction to data mining*. Pearson Education India.
- [Tangelder and Veltkamp, 2008] Tangelder, J. W. and Veltkamp, R. C. (2008). A survey of content based 3d shape retrieval methods. *Multimedia tools and applications*, 39(3):441–471.
- [Taubin, 1991] Taubin, G. (1991). Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138.
- [Tax and Duin, 1999] Tax, D. M. and Duin, R. P. (1999). Support vector domain description. *Pattern recognition letters*, 20(11):1191–1199.
- [Tax and Duin, 2004] Tax, D. M. and Duin, R. P. (2004). Support vector data description. *Machine learning*, 54(1):45–66.
- [Tejani et al., 2014] Tejani, A., Tang, D., Kouskouridas, R., and Kim, T.-K. (2014). Latent-class hough forests for 3d object detection and pose estimation. In *European Conference on Computer Vision*, pages 462–477. Springer.

- [Torr and Zisserman, 2000] Torr, P. H. and Zisserman, A. (2000). Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156.
- [Tuytelaars et al., 2011] Tuytelaars, T., Fritz, M., Saenko, K., and Darrell, T. (2011). The nbnn kernel. In *International Conference on Computer Vision (ICCV)*, pages 1824–1831. IEEE.
- [Tuytelaars and Mikolajczyk, 2008] Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280.
- [Ummenhofer and Brox, 2012] Ummenhofer, B. and Brox, T. (2012). *Dense 3d reconstruction with a hand-held camera*. Springer.
- [Viola et al., 2001] Viola, P., Jones, M., et al. (2001). Robust real-time object detection. *International journal of computer vision*, 4(34-47):4.
- [Wang et al., 2012] Wang, R., Shan, S., Chen, X., Dai, Q., and Gao, W. (2012). Manifold-manifold distance and its application to face recognition with image sets. *Image Processing (IP)*, 21(10):4466–4479.
- [Wolf and Shashua, 2003] Wolf, L. and Shashua, A. (2003). Learning over sets using kernel principal angles. *The Journal of Machine Learning Research (JMLR)*, 4:913–931.
- [Wu, 2011] Wu, C. (2011). Visualsfm: A visual structure from motion system.
- [Wu and Trivedi, 2008] Wu, J. and Trivedi, M. M. (2008). A two-stage head pose estimation framework and evaluation. *Pattern Recognition (PR)*, 41(3):1138–1158.
- [Yang et al., 2009] Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801. IEEE.

- [Yu et al., 2010] Yu, J., Zheng, H., Kulkarni, S. R., and Poor, H. V. (2010). Two-stage outlier elimination for robust curve and surface fitting. *EURASIP Journal on Advances in Signal Processing*, 2010:4.
- [Yuan and Shaw, 1995] Yuan, Y. and Shaw, M. J. (1995). Induction of fuzzy decision trees. *Fuzzy Sets and systems*, 69(2):125–139.
- [Zhang and Wang, 2006] Zhang, J. and Wang, H. (2006). Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and Information Systems*, 10(3):333–355.
- [Zhang et al., 2016] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.
- [Zhang et al., 2015] Zhang, Q., Wu, Y. N., and Zhu, S. (2015). Mining and-or graphs for graph matching and object discovery. pages 55–63.
- [Zoph et al., 2018] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. *computer vision and pattern recognition*.