Integrated Visual Perception Architecture For Robotic Clothes Perception and Manipulation

Li Sun

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Computing Science

College of Science and Engineering
University of Glasgow

March 2016

Abstract

This thesis proposes a generic visual perception architecture for robotic clothes perception and manipulation. This proposed architecture is fully integrated with a stereo vision system and a dual-arm robot and is able to perform a number of autonomous laundering tasks.

Clothes perception and manipulation is a novel research topic in robotics and has experienced rapid development in recent years. Compared to the task of perceiving and manipulating rigid objects, clothes perception and manipulation poses a greater challenge. This can be attributed to two reasons: firstly, deformable clothing requires precise (high-acutiy) visual perception and dexterous manipulation; secondly, as clothing approximates a non-rigid 2-manifold in 3-space, that can adopt a quasi-infinite configuration space, the potential variability in the appearance of clothing items makes them difficult to understand, identify uniquely, and interact with by machine.

From an applications perspective, and as part of EU CloPeMa project, the integrated visual perception architecture refines a pre-existing clothing manipulation pipeline by completing pre-wash clothes (category) sorting (using single-shot or interactive perception for garment categorisation and manipulation) and post-wash dual-arm flattening. To the best of the author's knowledge, as investigated in this thesis, the autonomous clothing perception and manipulation solutions presented here were first proposed and reported by the author. All of the reported robot demonstrations in this work follow a perception-manipulation methodology where visual and tactile feedback (in the form of surface wrinkledness captured by the high accuracy depth sensor i.e. CloPeMa stereo head or the predictive confidence modelled by Gaussian Processing) serve as the halting criteria in the flattening and sorting tasks, respectively.

From scientific perspective, the proposed visual perception architecture addresses the above challenges by parsing and grouping 3D clothing configurations hierarchically from low-level curvatures, through mid-level surface shape representations (providing topological descriptions and 3D texture representations), to high-level semantic structures and statistical descriptions. A range of visual features such as *Shape Index*, *Surface Topologies Analysis* and *Local Binary Patterns* have been adapted within this work to parse clothing surfaces and textures and several novel features have been devised, including *B-Spline Patches* with *Locality-Constrained Linear coding*, and *Topology Spatial Distance* to describe and quantify generic landmarks (wrinkles and folds). The essence of this proposed architecture comprises 3D generic surface parsing and interpretation, which is critical to underpinning a number of laundering tasks and has the potential to be extended to other rigid and non-rigid object perception and manipulation tasks.

The experimental results presented in this thesis demonstrate that: firstly, the proposed grasping approach achieves on-average 84.7% accuracy; secondly, the proposed flattening ap-

proach is able to flatten towels, t-shirts and pants (shorts) within 9 iterations on-average; thirdly, the proposed clothes recognition pipeline can recognise clothes categories from highly wrinkled configurations and advances the state-of-the-art by 36% in terms of classification accuracy, achieving an 83.2% true-positive classification rate when discriminating between five categories of clothes; finally the Gaussian Process based interactive perception approach exhibits a substantial improvement over single-shot perception. Accordingly, this thesis has advanced the state-of-the-art of robot clothes perception and manipulation.

Acknowledgements

First of all, I would like to thank my supervisor Dr. Paul Siebert, who is nice, caring and full of knowledge. He gave me a ticket to study in the UK and introduced me to the world of computer vision and robotics. Without him, I would have been frozen to death in Quebec where I had planned to study. I might have also died of the desperation for not knowing a single word of French. In a word, Paul has saved my life. Even more to his credit, he has changed my whole life by letting me to work on a EU project in which I could play with the cutting-edge robots. He opened my mind to robot vision which is the area I am going to pursue for my life.

Second, I would like to thank my second supervisor, Dr. Simon Rogers, who has given me much more help than most second supervisors would do. He is an expert of machine learning with rigorous standards. I have learnt a lot in machine learning from him.

Third, I would like to thank Prof. Vaclav Hlavac for approving my work and providing reference for me. I like his way of working and am looking forward to the chance to collaborate with him in the future.

Fourth, I am grateful for my colleagues and friends. A big brother for me, Gerardo,is good at all nasty technical problems. I learnt many technical skills from him and he has given me lots of life advice apart from research. As to Aamir, he is a steady friend who has always been there for me during my hardest time. I cannot say thanks enough to him. My thanks also goes to Yuan, Tian, and other friends in our school.

In addition, I am indebted to my friend Partick who has proofread my thesis. Also, I would like to thank the friends I met in CloPeMa for their generous help and sharing during the project. They are Jane, Validmir, Libor in CVUT and Andrew in CERCH.

Moreover, I would like to thank my girlfriend Luna. Without her, I could have completed this thesis three moths ago and had at least another two good publications during my PhD. But I never regret to meet and fall in love with this lovely girl. The time with her is my most precious memory in the UK.

Finally, I would like to thank my parents for their unconditional love and they are always the shield against all the difficulties of my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The main aim of this research is to advance the state-of-the-art in robotic clothes perception and manipulation. Robotic clothes manipulation is a novel research topic and has been attracting increasing attention in recent years. This is a challenging topic compared to rigid object manipulation since more precise perception, more invariant interpretation and more dexterous manipulation skills are required. This can be achieved by parsing or interpreting the underlying shape information from 3D clothing surface in order to understand the clothing's configuration. This proposed method is underpinned by the generic 3D surface analysis in which the 3D shape and topologies comprising both types and quantifications can be obtained at local 3D surface. Moreover, inspired by a human vision based manipulation mechanism, the perception-manipulation cycles are employed in integrated autonomous robot systems, achieving fully-autonomous visually-guided clothes manipulations and recognitions with halting-control, multiple feedbacks guidance.

1.1 Aims and Objectives

From the application perspective, autonomous laundering is the main targeted application of robotic clothes perception and manipulation. A previously existed and completed laundering pipeline is shown in Fig. 1.1, comprising three stages: pre-washing stage, washing stage and post-washing stage. During pre-washing stage, the clothes should be classified and sorted with respect to their materials or categories. Then, the clothes are washed separately. After being washed, the washed and dried clothes are stacked in a pile. The robot or manipulator grasps one of them and then unfolds, flattens and folds it. The flattening and folding procedures are conducted on an operating table. This process is repeated until all the clothes in the pile are folded.

Current research has advocated to solve subtasks within an autonomous laundering pipeline,

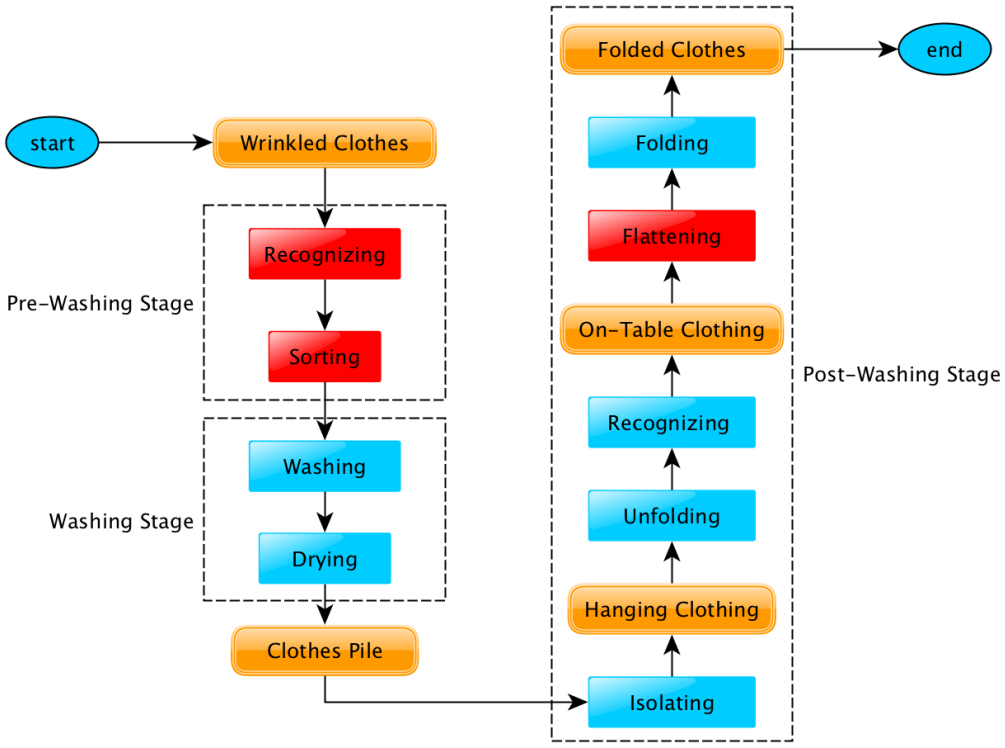


Figure 1.1: The autonomous laundering pipeline. In this figure, rectangles indicate the process and rounded rectangles refer to the clothing. More specifically, the blue blocks refer to the existing processes proposed by previous work, while the red ones are new processes proposed by this thesis, and yellow blocks indicate the status of clothing.

for instance: grasping clothes from a heap of garments [Ramisa et al., 2012, **?**], recognising clothes categories [Ramisa et al., 2012, Willimon et al., 2013a, 2011a], unfolding [Cusumano-Towner et al., 2011, Doumanoglou et al., 2014a,b, Li et al., 2015a, Willimon et al., 2011a], garment pose estimation [Kita et al., 2009a,b, Li et al., 2014a,b] and then folding [Li et al., 2015b, Maitin-Shepard et al., 2010, Miller et al., 2012, Stria et al., 2014b, Van Den Berg et al., 2011].

In previous research, the pre-washing stage is neglected. Whereas, in real life, washing clothes of different categories together entails a health hazard, and doing so with a fixed washing program may cause permanent damage to clothes of vulnerable materials. In recent years, these problems have attracted increasing attention from both customers and white-goods companies. Furthermore, the previously proposed laundering pipeline also misses the flattening process. After being unfolded, the garment is placed on the operating-table. Although the sliding-table-edge strategy is used to flatten the garment, the wrinkles cannot be completely removed and will be carried into the folding procedure. This thesis refines the autonomous laundering pipeline by completing the pre-washing stage and improving the flattening procedure in the post-washing stage.

From a scientific perspective, although the proposed research is applied to robotic clothes manipulation, the objectives of this research focus on clothing but are not constrained to clothing. Firstly, this research aims to explore the means of manipulating highly deformable objects. Secondly, this research aims to investigate the robust visual interpretation of objects of quasi-infinite configuration space. Finally, this research aims to investigate an advanced visually-guided robot and object interaction routine, which can be extended to rigid and non-rigid objects manipulation.

1.2 Scientific Motivations

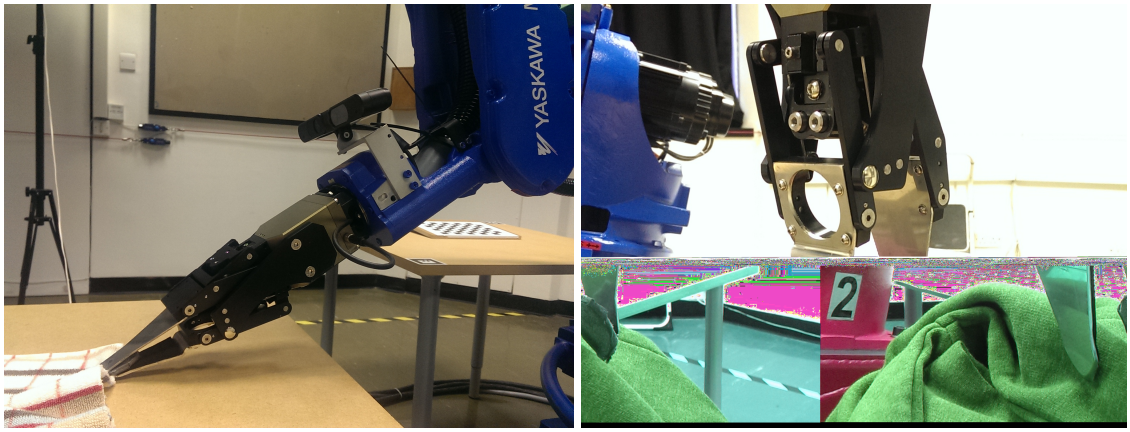
The general idea is to 'parse' and 'interpret' the 3D configuration of the clothing. Clothes are of various pattens, textures, and colors which are unstable characteristics of garments themselves. Compared to parsing/recognising clothes from RGB images, the 3D configuration observed from 3D depth map/point cloud is more robust to changing of a clothing's configurations. Therefore, this proposed visually-guided clothes manipulation and recognition approach is based on understanding the 3D clothing configuration by the means of parsing and interpreting 3D clothes landmarks.

The 3D clothing configuration is comprised of generic clothing surface and its landmarks. The landmarks consist of specific landmarks e.g. pocket, collar, sleeve, cuff, etc. and generic landmarks e.g. wrinkles, folds, 3D clothes textures, etc, in which generic landmarks are more robust and generic information. In this thesis, for parsing and interpreting the cloth, usually only generic landmarks are considered because these are not constrained to specific categories and unlikely to be susceptible to deformable configurations and occlusions.

Now the roles of landmarks in clothes perception and manipulation are illustrated. In the visually-guided clothes manipulation task such as grasping and flattening, the intuition is to detect and parametrise the landmarks (mainly are wrinkles) in order to guide the robot to localize the position and orientation for grasping and flattening. In clothes category recognition tasks, the intuition is to describe the 3D clothing configuration from the statistics of 3D shape of landmarks, surface shape types, topologies measurements and 3D textures because these are robust indicators and diagnostics of clothes categories.

1.3 Challenges

In robotic clothes perception and manipulation, there are two general challenging problems this thesis intends to resolve.



(a) A demonstration of picking garment from the table.

(b) A demonstration of grasping garment from above.

Figure 1.2: Examples of different types of dexterous manipulation for clothing.

The physical form of cloth is between solid and liquid depending on the characteristic of the fibre and manufacturing processes. Due to this special physical form, robotic clothes manipulation requires dexterous manipulation skills and specially-designed grippers. In Fig. 1.2, the two most commonly used types of manipulation skills are shown. They are: picking clothing from the table by grasping the clothing's edge and grasping clothing from the above by fetching the clothing's wrinkle. Both of them require dexterous manipulation. The robot vision system should be able to provide accurate perceptions in order to conduct dexterous manipulations. Therefore, how to acquire accurate 3D sensing and parse the garment's configurations precisely are the critical problems that need to be solved in visually-guided clothes manipulation.

(a) Samples from the configurations of a cup.

(b) Samples from the configurations of a t-shirt.

Figure 1.3: The comparison between rigid objects and non-rigid objects with respect to the configuration space.

In this research, the configurations of clothing refer to the 3D manifold of 2D surface, while in robot vision system, they are observed as 3D deformable surfaces because of visual occlusions. These clothing surfaces are physically continuous under the constraints of clothing's

fibres. Geometrically, the clothing surfaces are treated as C1 continuous (the changing of first order derivatives is constant)[1]. Since the clothes surfaces are highly deformable, the configuration space generated is infinite. An example of comparisons between rigid objects and non-rigid objects on configuration space is given in Fig. 1.3. In the 3D space, the clothing configuration can be described as a 3D implicit surface in which the outside layer is captured but inside is occluded. As most clothes are textured, the physical clothes configuration can be rendered by various colours and patterns. The quasi-infinite configuration space, occlusions and various patterns made visual perception of clothes extremely challenging. How to devise robust and invariant visual representation to clothing's deformable form is of significant importance to the manipulation and recognition tasks.

From the two problems presented above, the key difficulties of robotic clothes perception and manipulation can be concluded as follows:

- Clothing is practically the most deformable non-liquid object in real life, and handling a piece of clothing requires dexterous manipulation.
- In order to conduct dexterous visually-guided manipulation, accurate depth sensing is necessary, which can capture the tiny landmarks that are the diagnostic of a garment's structure. Visual perception mechanisms that can detect and quantify these landmarks precisely are also required.
- The 3D configurations generated as the manifold of 2D garment surfaces are of quasi-infinite space. It is extremely difficult to interpolate this infinite configuration space from limited amount of training examples.
- Due to the huge configuration space, some configurations are ill-posed, and it is difficult to recognizing unknown garments from these ill-posed configurations.

1.4 Hypotheses

The objective of this thesis is to advance the state-of-the-art of robotic clothes manipulation, especially on visually-guided manipulation and clothes category recognition, through a full-understanding and an enriched interpretation of garment 3D configuration. Inspired by the mechanism of mammalian brains, in this thesis, the integrated autonomous robot systems are devised following perception-manipulation cycles. More specifically, the hypothesises of this thesis are three-fold:

[1] B-Spline Surface Fitting is employed to enforce the surface continuity, which is introduced in Appendix A

- In order to manipulate a garment, the 3D garments structures can be identified for the grasping or flattening purpose if the garment's local surface shapes are sufficiently understood. In addition, metric information specifies the dimensions of these structures must also be recovered through vision in order to determine size compatibility with the end effector being used to manipulation these structures.

- The category of a garment can be recognised from any free-configuration if a robust interpretation that is invariant to the garment's deformable form is proposed. 3D based descriptions of clothing configuration are more robust than RGB based representations for relatively small scale dataset.

- By employing multiple perception-manipulation cycles, both robotic perception and manipulation goals can be incrementally approached in the integrated autonomous robot system (this process can be non-monotonic).

1.4.1 Key Issues and Potential Solutions

Corresponding to the hypothesis presented above, there are three key issues that need to be addressed in this thesis:

- What are required for a sufficient understanding of 3D clothes configuration?
  A sufficient understanding aims to recognise, localise and quantify the clothes landmarks on the 3D clothes surface. In order the achieve this, a full understanding of the 3D shape and topologies are required by which the generic landmarks e.g. grasping atoms or wrinkles, can be detected and precisely localised through their geometric definition. In addition, quantification of the landmarks, ranking the priorities of the detected landmarks and indicating the magnitude of continuous action, e.g. pulling distance in flattening, are also of significant importance.

- What is a robust interpretation of 3D clothes configuration?
  Interpretation of 3D clothes serves to instantiate the high-dimensional configurations into high-level descriptions that can be used for learning purposes. For each item of clothing, it is of infinite variant configurations; for each category of clothes, intra-variance exists between different items of clothings; for all types of garments, extra-variance exists between different categories of clothings. Corresponding to these three kinds of variances, a robust clothes interpretation should be able to interpolate the quasi-infinite configurations from limited training examples by means of minimizing the intra-dissimilarity with the each category and maximizing the extra-dissimilarity among different categories.

- How to set the halting criteria in perception-manipulation cycles?
  The halting criteria determine when to terminate the interactive-perception routine. Since the perception-manipulation aims to incrementally approach the manipulation or perception goal, the halting criteria should be designed to target when that goal is completed. In autonomous robot systems proposed in this thesis, the statuses of tasks are tracked through visual and tactile feedbacks. To be more specific, the halting criteria of different tasks are different: in autonomous grasping, tactile feedback indicating the gripper is holding the garment is set as the halting criteria; in garment flattening, the visual feedback showing that the garment's wrinkledness is reasonably low or the captured wrinkle is too tiny to flatten is used as the halting criteria; in garment recognition, the confidence of the prediction is devised as the halting criteria.

1.5 A Brief View of the Proposed Approach

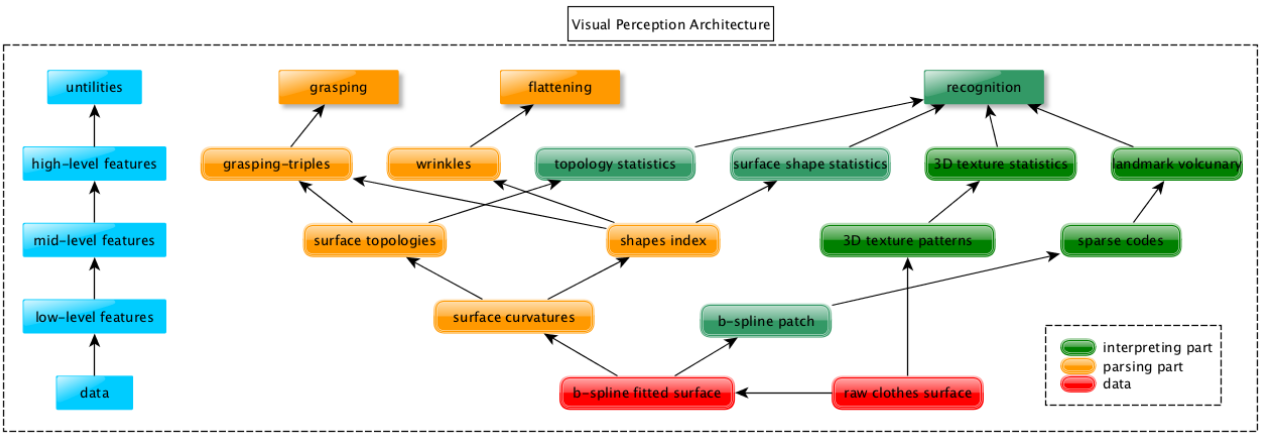


Figure 1.4: The hierarchical visual perception architecture for clothes perception and manipulation.

The proposed visual architecture consists of two aspects: the 'parsing' part, i.e. parsing the 3D clothes configurations hierarchically, estimating low-level surface curvatures, yielding surface shape and topology maps, and precisely localising and quantifying the generic landmarks for the purposes of grasping and flattening; the 'interpretation' part, i.e. interpreting the 3D clothes configurations by the means of fusing local landmark codes and the statistics of 3D texture, 3D shape types and 3D topologies of the generic clothes surface.

In Fig. 1.4, the 'parsing' part of the visual architecture is marked as yellow, the 'interpretation' part is marked as green, while red blocks indicate raw 3D data and pre-processed 3D data. The first part parses the the generic landmarks (wrinkles) of a 3D garment surface by the means of surface shape and topology analysis. As a consequence, the geometrical structure of wrinkles are precisely detected and quantified, by which the dexterous manipulations

such as grasping and flattening are guided. The second part interprets 3D generic landmarks (e.g. wrinkles and folds) through the vocabulary representation of local 3D descriptor; The 3D textures of the garment are interpreted through calculating the statistics of local binary patterns (LBP); The 3D shape types of garment are interpreted through the statistics of the shape classification of geometry surface (shape index); The 3D topologies of the garment are interpreted through statistics of the magnitude of spatial distances between different topology lines.

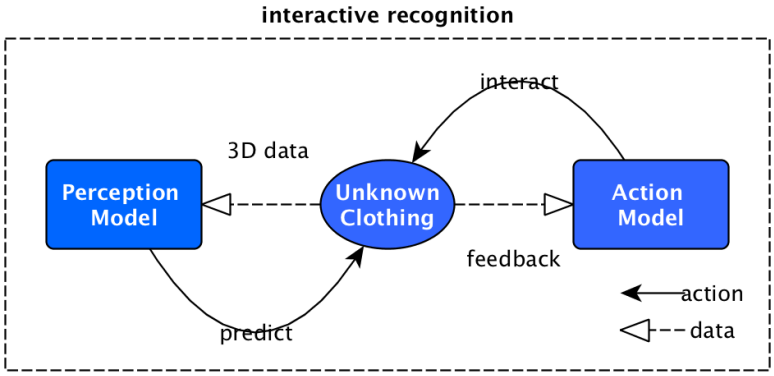


Figure 1.5: The model of perception and manipulation cycle.

Moreover, in order to approach the perception and manipulation goals incrementally, the proposed autonomous robot systems employ an perception-manipulation methodology. As shown in Fig. 1.5, in each cycle, the robot starts with perception, parsing the 3D configurations or interpreting the observed garment and predicting its category. Then consequently the feedbacks e.g. tactile feedback, visual feedback of 'wrinkledness' and predictive confidence, are received which decide whether the current state meets the halting criteria, then the action model drives the robot to interact with the garment after the receiving the guidance from the perception model.

1.6 The Impact of this Research

The scientific impact of this thesis are two-fold. Firstly, this thesis refines the existing pipeline by including the procedures of recognizing, sorting and flattening wrinkled clothes, which are usually neglected in those most well-known pipelines. Secondly, this thesis solves or advances two significant problems in robotic clothes perception and manipulation: guiding a robot to conduct dexterous operations and recognizing deformable objects of quasi-infinite configuration space.

More specifically, the scientific contributions of this thesis can be summarised as follows:

- A generic visual perception architecture towards fully-understanding 3D configuration of deformable clothing which is adapted to multiple robotic tasks: clothes grasping, flattening and clothes category recognition.
- The first autonomous dual-arm flattening solution for robot laundering. This approach follows a perception-interaction loop, and there are no constraints on the shape of garment.
- The first autonomous clothes category sorting solution for pre-washing procedure of robot laundering. This approach is based on a novel 3D interpretation with enriched visual features.
- The first probabilistic classification based interactive sorting solution which can remarkably improve the recognition performance.
- The first high-resolution (16 Megapixel) large-scale RGB-D dataset for clothes category recognition. The whole dataset and toolbox are available online: https://sites.google.com/site/clopemaclothesdataset/

1.6.1 Robot Demonstration

- Dual-arm garment flattening: https://www.youtube.com/watch?v=Z85bW6QqdMI
- Single-shot clothes sorting: https://www.youtube.com/watch?v=woQVqtmQc4M
- Interactive clothes sorting: https://www.youtube.com/watch?v=zsmrcqsTPGQ

1.6.2 List of Publications

- Li Sun, Simon Rogers, Gerarado Aragon-Camarasa, J. Paul Siebert. "Recognising the Clothing Categories from Free-Configuration using Gaussian-Process-Based Interactive Perception", accepted by ICRA 2016.
- Li Sun, Gerardo Aragon-Camarasa, Paul Siebert, Simon Rogers. "Accurate Garment Manipulation using Binocular Stereo Head with the Application to Dual-Arm Flattening", ICRA 2015. *Best Conference Paper*, *Best Student Paper*, *Best Manipulation Paper* Nominations.
- Li Sun, Gerarado Aragon-Camarasa, Simon Rogers, J. Paul Siebert, Aamir Hkan. "A Precise Method for Cloth Configuration Parsing Applied to Single-Arm Flattening", accepted by International Journal of Advanced Robotics Systems.

- Adam Schmidt, Li Sun, Gerardo Aragon-Camarasa, J. Paul Siebert. "The Calibration of the Pan-Tilt Units for the Active Stereo Head". Image Processing and Communications Challenges 7, 389, pp. 213-221.

- Li Sun, Gerardo Aragon-Camarasa, Paul Siebert, Simon Rogers. "Advances in robot manipulation of Clothes and Flexible Objects", ICRA Workshop on Clothes Manipulation. 2014.

- Li Sun, Gerarado Aragon-Camarasa, J. Paul Siebert, Simon Rogers. "A Heuristic-Based Approach for Flattening Wrinkled Clothes", 14th Towards Autonomous Robotic Systems, 2013.

  Submitted Paper:

- Li Sun, Gerarado Aragon-Camarasa, Simon Rogers, J. Paul Siebert. "An Integrated Visual Perception Architecture for Visually-Guided Clothes Manipulation", submitted to IEEE Transaction on Robotics.

1.7 The Structure of this Thesis

In Chapter 2, the background and provide a comprehensive literature review are presented. The background will introduce the CloPeMa robot and the stereo vision system, and the literature review will cover depth sensing, depth data analysis, object recognition, and the state-of-the-art of robotic clothes perception and manipulation. Then the achievements of this thesis are presented in the following four chapters. The first aspect of achievements, namely visually-guided clothes manipulation will be presented in Chapter 3 and Chapter 4. The second aspect, the clothes category recognition work regarding highly-wrinkled configurations, will be detailed in Chapter 5 and Chapter 6. To be more specific, in Chapter 3, the visual architecture for manipulation will be introduced and verified in physical simulation. The experiments demonstrate that a more comprehensive and precise configuration understanding is able to advance the manipulation (flattening) performance. In Chapter 4, the proposed visual architecture is integrated into CloPeMa dual-arm robot testbed and both grasping and flattening are demonstrated and evaluated. Chapter 5 presents the other part of the proposed visual architecture (representation part) while Gaussian Process based interactive perception is illustrated in Chapter 6. Finally, the conclusion of the whole thesis and the future work are given in Chapter 7.

Chapter 2

Literature Review

This chapter provides a comprehensive review of previously reported theories, methodologies and applications which are relevant to the proposed research. Developing a novel visual architecture and integrating it into autonomous robotic systems requires a wide range of research regarding hardware and software architectures, computer vision and robotics skills. This chapter can be divided into three parts: background, computer vision literature review and robotics literature review. The background presents the related hardware and software knowledge used in the robot testbed. In the computer vision literature, the proposed visual architecture includes two parts, parsing and interpretation, and so the 3D data analysis and object recognition techniques are reviewed. In the robotics literature, both rigid object and non-rigid object manipulations are reviewed. The advantages and shortfalls of the reviewed approaches are discussed, and suggestions for ways to advance the-state-of-the-art is included at the end. A more detailed structure is as follows.

In section 2.1, the background knowledge is presented which is necessary for having a better understanding of the proposed methods reported in the rest of this thesis. It includes the mechanical design of CloPeMa robot, the software packages used to control the robot and the stereo matching algorithm for depth sensing. Section 2.2 investigates the widely-cited depth sensors and sensing approaches in robot manipulation. Then, the generic 2.5D or 3D data analysis approaches are reviewed in Section 2.3. Since this thesis includes clothes recognition tasks, the general object recognition pipeline is reviewed in Section 2.4. Before clothes manipulation tasks are introduced, the related robot manipulation work on rigid objects is probed in section 2.5. Finally, Section 2.6 presents the state-of-the-art achievements in robot clothes perception and manipulation.

2.1 Background

2.1.1 The CloPeMa Project

The CloPeMa ("Clothes Perception and Manipulation", www.clopema.eu) is a collaborative EU FP7 project involving University of Glasgow, CRTH[1], CVUT[2] and University of Geneva[3]. CloPeMa aims at advancing the state of the art in the artificial perception and dexterous manipulation of clothes and other textiles.

During the project (from 2012.2 to 2015.2), tactile sensing, visual sensing and soft materials manipulation were jointly managed by a goal driven, high-level reasoning module. Inspired by the perception-manipulation cycle of the mammalian brain, the reasoning module also provided perception capabilities to fuse sensing and manipulation. The task calls for hierarchical representations and related perception-manipulation skills of different complexities. These theories addressed real-life autonomous laundering problems, e.g. dual-arm garment folding [Stria et al., 2014b], unfolding [Doumanoglou et al., 2014a,b], dual-arm flattening [Sun et al., 2015], interactive sorting [Sun et al., 2016] and a novel gripper design [Thuy-Hong-Loan Le et al., 2013].

2.1.2 The CloPeMa Robot

Overall

The CloPeMa robot is an off-the-shelf robot that is specially designed for clothes perception and manipulation. The robot is shown in Fig. 2.1. The robot is comprised of the main robot body, robot head, robot grippers and other sensors, which are introduced in the following sections.

Robot Body

Robot clothes manipulation requires a dexterous, reliable dual-arm robot that is able to handle adult clothing. Before CloPeMa, the available off-the-shelf dual-arm robots such as PR2 were too small to manipulate adult clothing. This was the motivation for CloPeMa to produce its off-the-shelf dual-arm robot.

The main robot body is based on the industrial robotic components for welding operation which are supplied by YASKAWA Motoman. As shown in Fig. 2.1, two MA1400 manipulators are used as two robot arms. Each manipulator is of 6 DOF, 4 kg maximal load weight,

[1]Center for Research and Technology Hellas, http://www.certh.gr/root.en.aspx
[2]Ceske Vysoke Uceni Technicke V Praze, https://www.cvut.cz/en?set_language=en
[3]Universita Degli Studi Di Genova, https://unige.it/

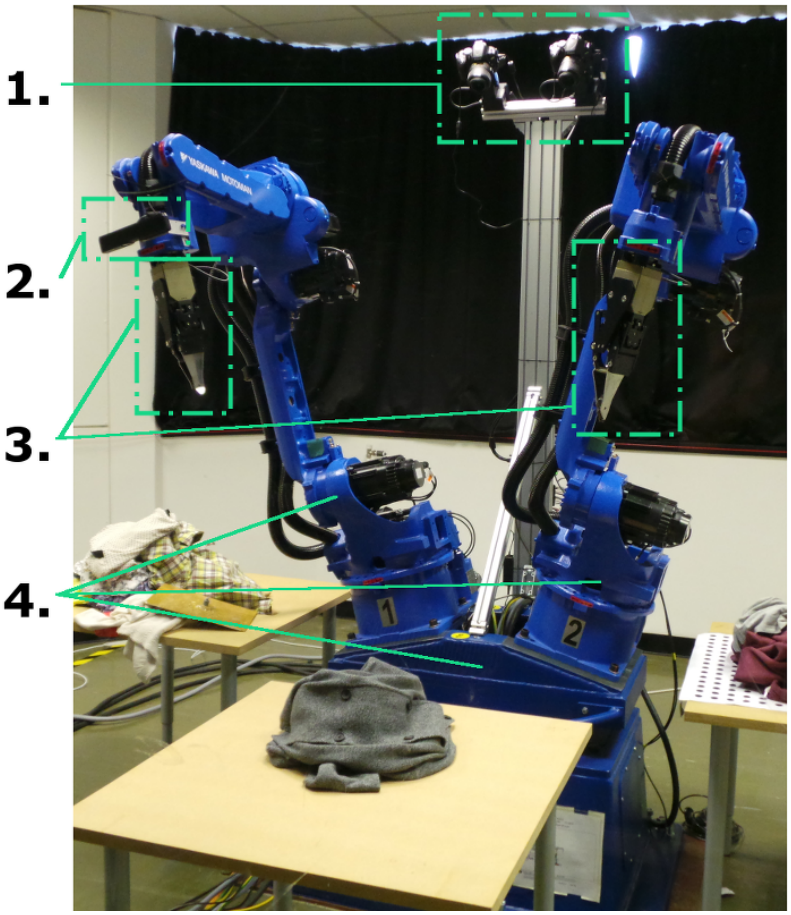


Figure 2.1: The CloPeMa robot.

1434 mm maximal reaching distance, ±0.08 mm accuracy. These specifications satisfy the requirements for conducting accurate adult clothing manipulation. They are mounted on rotatable turning tables. The robot arms and turning table are powered and controlled by DX100 controllers.

Robot Head

Differing from most of the state-of-the-art projects, CloPeMa aims to use relatively inexpensive, commercially available component elements to build an off-the-shelf robot vision system (binocular head) for depth sensing. The goal of this is to offset the limitation of widely-used depth sensor such as Kinect with respect to accuracy and resolution.

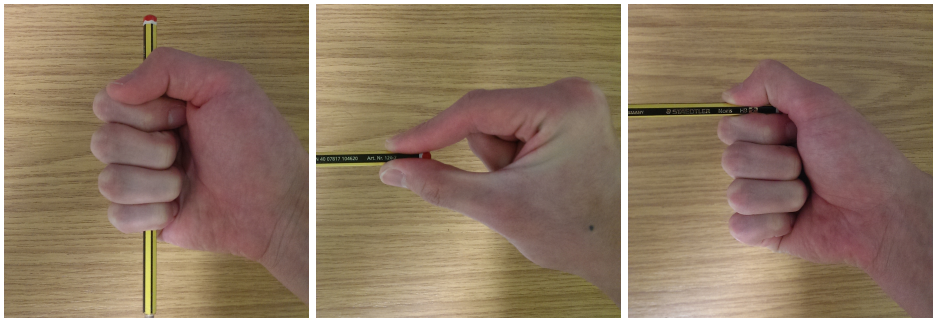
The robot head comprises two Nikon DSLR cameras (D5100) that are able to capture images of 16 mega pixels through USB control. Gphoto library[4] is employed to drive the capturing under ubuntu. These are mounted on two pan and tilt units (PTU-D46) with their corresponding controllers. The cameras are separated by a pre-defined baseline for optimal stereo

[4]http://gphoto.sourceforge.net/

capturing. Its field of view covers the robot work-space. The robot head provides the robot system with high resolution 3D point cloud.

Robot Grippers

The mechanical design of the CloPeMa gripper is inspired by the way humans grip. In the book 'Examination of the Hand and Wrist' [Tubiana et al., 1998], human grips have been categorised into power grip (digitopalmar grip), precision grip (thumb-finger pinch) and lateral pinch. The robot's dexterous manipulation on clothes is more likely to imitate the human's precision grip. This is because it can enable the robot to pinch a small component or edges of a garment. For this purpose, the CloPeMa gripper is designed as two thin fingers powered by liquid-pressure and with tactile sensors integrated on the tips.



(a) A demonstration of the power grip. (b) A demonstration of the thumb-finger pinch. (c) A demonstration of the lateral pinch.

Figure 2.2: Three types of human grip.

The University of Genoa (UNIGE) has developed CloPeMa prototype grippers for grasping clothes based on Schunk grippers. The prototype has a tactile sensor at the "finger tips" to sense the garment material using little rubbing motions between the "gripper fingers". Fitted with the multi-modal tactile sensor it can also detect proximity, sound and pressure in order to sense buttons and other smaller details of clothing. Furthermore the gripper has patented variable stiffness actuators for adapting to different grasping and steering tasks.

Other Sensors

The robot has been fitted with two to three Xtion Pro Live from ASUS for light weight range sensing, one on each arm and one on the back spine. These are used for depth sensing.

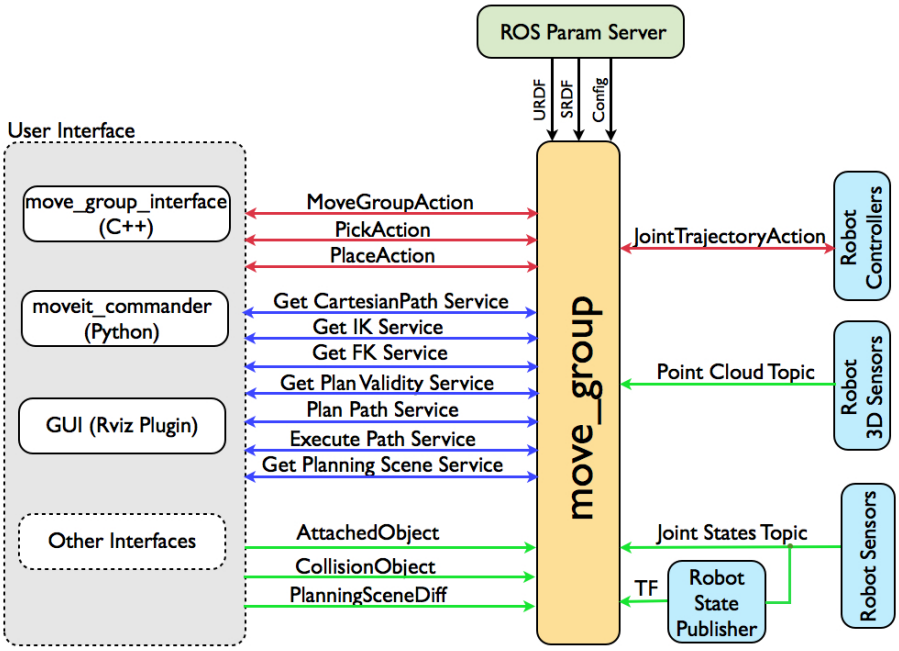


Figure 2.3: The working schema of Moveit.

2.1.3 Robot Control

The ClopeMa robot is fully integrated with Robot Operating System (ROS) through ROS industrial package[5]. The architecture of the robot control system is presented in Fig. 2.3. More specifically, the URDF (uniform robot description form) is used to define the geometric structure of the robot. After the geometric structure is defined, collision can be detected by robot collision models, and the transforms between robot links can be achieved by TF[6]. MoveIt package is employed to achieve the communication between user interface and robot controllers.

2.1.4 Stereo Head Calibration

The stereo head calibration has two steps: camera calibration and hand-eye calibration. The former is to estimate the intrinsic parameters of the stereo cameras in order to reconstruct 3D scene from the disparities. For the CloPeMa robot, the OpenCV calibration routines[7] are employed to estimate the intrinsic camera parameters of each camera. Furthermore, hand-eye calibration is employed to link the unknown stereo head into robot coordinate. In other words, the unknown transformation from the camera frame to the calibration pattern coordinate system, as well as the transformation from the calibration pattern coordinate system to

[5] http://wiki.ros.org/Industrial
[6] http://wiki.ros.org/tf
[7] http//opencv.org

the hand coordinate system, need to be estimated simultaneously. For the CloPeMa stereo head, Tsai's hand-eye calibration [Tsai and Lenz, 1988, 1989] routines are used to estimate rigid geometric transformations between camera to chess board and chess board to the gripper.

2.1.5 Stereo Matcher

Stereo matching is the key procedure of depth sensing of CloPeMa robot head. The objective of stereo matching is to ascertain the correspondence of the pair of images. The correspondence is specified by disparity maps in horizontal and vertical directions respectively. The stereo matching approach integrated in CloPeMa stereo head is principally based on C3D [Siebert and Urquhart, 1995, Zhengping, 1988]. C3D is a local-correlation matcher working on scale space. More specifically, after the Gaussian pyramid is built, for every pixel in the left image, the correlation value is calculated and matched with its neighbouring pixels in the right image:

$$cor_{h,v}(x,y) = \sum_{u}^{l}\sum_{v}^{r} I_l(x+u, y+v) I_r(x+u, y+v) w(u,v) \tag{2.1}$$

Image correlation is weighted by a Gaussian kernel which is of size $l \times r$, and $w(u,v)$ defines the weight of the Gaussian kernel within the window. As shown in Fig. 2.4, the correlation procedure is performed from coarse to fine scales in the Gaussian pyramid, and the correlation on the finer scale depends on the disparities estimated by its coarser scale. The final disparity maps can be produced after the matching is completed iteratively for all the levels of the pyramid.

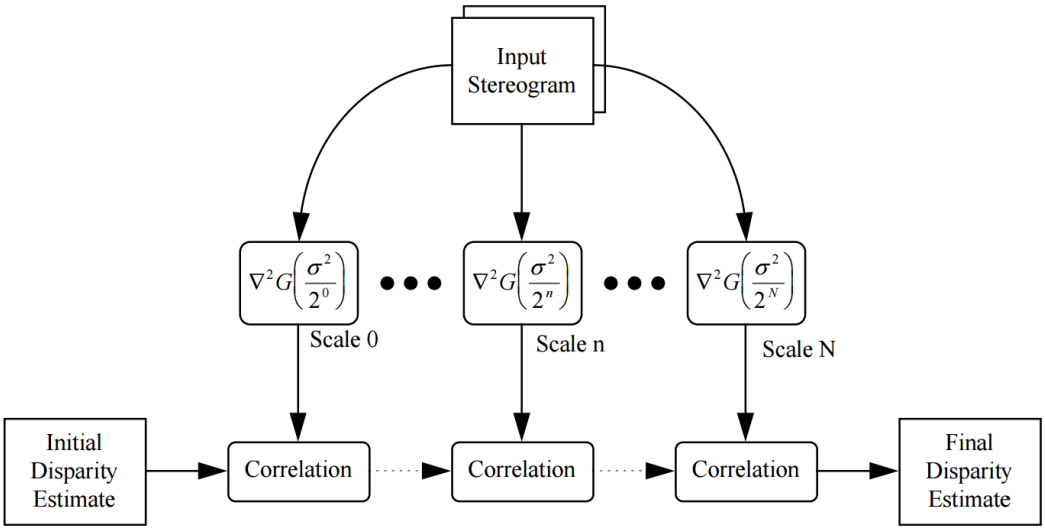


Figure 2.4: The working framework of C3D matcher.

Differing from most of the existing stereo matching approaches, the two images are not rectified, therefore the disparities in both horizontal and vertical direction are calculated. Also, the displacement can be non-integer values as a result of applying a bi-linear interpolation on the correlation values.

2.1.6 Clothes Attributes

From Fibre to Fabric to Clothing

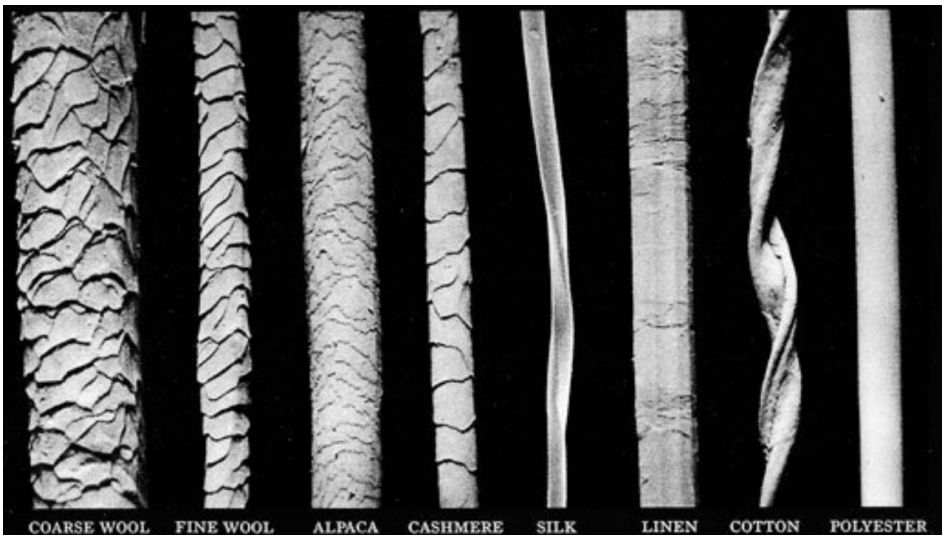


Figure 2.5: The examples of different fibres under microscope.

Cloth (or Fabric) is made from different textile manufacturing processes such as weaving, felting, knitting etc. of fibres. Fibres consist of natural fibres and synthetic fibres. The commonly-used natural fibres are silk, cotton, wool, flax etc. and the synthetic fibres include polyester, acrylic, poly-amides, polypropylene and polyurethane. Examples of widely-used fibres are shown in Fig. 2.5. The general 'material' of clothes is determined by both its fibres and manufacturing process. The clothing is made of fabric by tailoring and sewing. The automation of fabric manufacturing has been achieved for hundreds of years, whereas automatic or autonomous clothing manufacturing (e.g. t-shirt, shirts, jeans etc.) is still underdeveloped.

Physical Clothes Simulation

For clothes simulations, the most commonly cited model for the physical structure of cloth surface is the mass-spring model [Provot, 1995], in which clothes particles are constrained by three types of constraints: structural spring, shear spring and flexion spring (as shown in Fig.

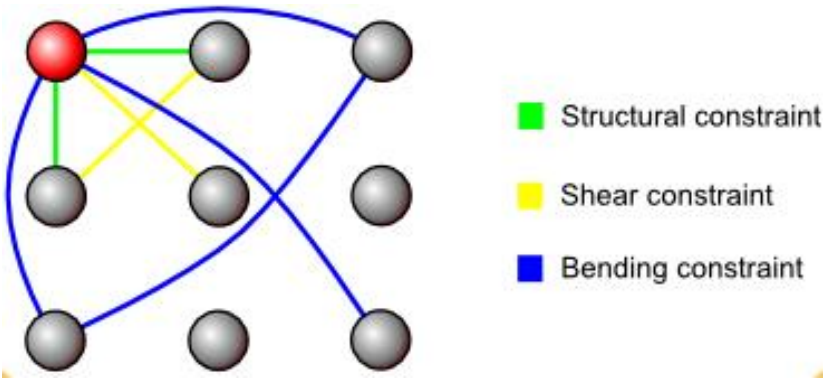


Figure 2.6: The mass-spring model of cloth.

2.6). These springs provide different connections and resistance between neighbouring particles. Structural springs connect the mesh in a rectangular grid. They provide the structural basis for simulated cloth. Shear springs provide a resistance force to shearing. They connect the points diagonally. Bend springs provide resistance to the cloth from simply folding on itself. They are connected between every other point. More generally, these constraints can be quantified into several material attributes: thickness, stiffness and bendiness.

Physical clothes simulation is of significant importance in clothes perception and manipulation. Firstly, in real-life, it is impossible to create completely the same clothes configurations, while in simulation, the clothes configuration can be duplicated by recording the positions and constraints of particles. This characteristic is utilized to evaluate the performance of visually-guided manipulations by duplicating the same configurations and conducting interactions guided by different visual-perception approaches. Secondly, physical clothes can generate mass synthetic data, and in some research [Li et al., 2014a,b, 2015a,b], the recognition or manipulation models are trained by simulated data and tested on real clothes. Thirdly, in some reported research [Kita et al., 2004, 2009a,b, Schulman et al., 2013], clothes categories or poses are estimated by registering simulated clothes with the point cloud of real clothes.

Clothes Landmarks

Landmarks usually are salient structures in a dynamic environment. They are also the robust structures in a deformable object. Take a human face for an example: The nose, eyes, ears and mouth are the landmarks of a deformable face. In deformable clothes, visual landmarks can be divided into specific landmarks and generic landmarks. The specific landmarks e.g. pocket, collar, sleeve, cuff etc., are constrained by clothes categories and are not distributed among all categories. The generic landmarks including wrinkles, folds and 3D clothes textures are distributed on the clothes surfaces among all clothes categories. Inspired by face recognition work, landmarks are used to describe clothes configuration. In the proposed

approach, both specific and generic landmarks are likely to be captured, although generic landmarks play a more important role in the representation.

2.2 Depth Sensing for Robot Manipulation

In this section, a survey of widely-cited depth sensing methods is presented. The stereo-matching based depth sensing and the Kinect-like cameras are investigated respectively.

2.2.1 Stereo-Matching Based Binocular Cameras

Before Kinect-like cameras were available, stereo-vision based depth sensing had been widely investigated. The stereo 3D reconstruction pipeline consists of: (1) capturing synchronised pair images from stereo cameras, (2) finding correspondences between pixels from the pair images in order to obtain disparity maps (i.e. stereo matching) and (3) reconstructing 3D scene from the disparities maps using calibrated camera parameters of the stereo system. Within the pipeline, stereo matching is a critical step to recover accurate 3D maps which is also computationally expensive. Below, a brief review of the two main categories of stereo matching algorithms is given, namely local and global methods, and described the motivation for the stereo matcher used in this thesis.

The goal of stereo-matching is to reduce the disparity space image (DSI) error, which is the difference between corresponding pixels after applying disparities. In order to measure DSI error, different correlation measurments are proposed, such as square intensity differences [Hannah, 1974], absolute intensity difference [Scharstein and Szeliski, 2002], and other speific corrlation measurement [Hirschmüller and Scharstein, 2009]. Local method is devised to reduce the DSI in local regions (can be pixels or patches) and iterative-strategies are widely used to reduce the DSI error. In [Bergen et al., 1992, Quam and Center, 1984, Siebert and Urquhart, 1995, Witkin et al., 1987], a coarse-to-fine correlation searching on scale pyramid is proposed in order to achieve a more efficient matching. Within local approaches there are non-correlation-based methods such as finding the correspondences through multi-scale image edges [Marr and Poggio, 1979, Marr and Vision, 1982]. Global approaches formulate the stereo matching problem as minimization of the energy function incorporating the DSI error term and a smoothness term. The smoothness term is usually modelled by discontinuity-preserving functions [Black and Rangarajan, 1996] as the difference of disparities among adjacent pixels. In order to solve this optimization problem, the most widely-cited approaches are: dynamic programming-based method [Baker, 1982, Ohta and Kanade, 1985] and the segmentation-based methods [Scharstein and Szeliski, 2002]. It

is worth noting that the latter methods are more effective in handling scenes of discontinuous depth.

2.2.2 Kinect-Like Cameras

The most famous on-the-shelf depth sensors are Kinect-like cameras (including Mircosoft Kinect and Asus Xtion Pro), which have been widely used in RGB-D based recognitions. The sensing range of Kinect-like cameras is 0.8m to 3.5m and sensing precision is approximately 5mm. Kinect achieves real-time depth sensing (30FPS), but for each image frame the quality of depth map is limited. Afterwards, Kinect-fusion [Izadi et al., 2011, Newcombe et al., 2011] is proposed for high-accuracy 3D reconstruction of static objects or indoor scenes by merging multiple frames from a moving kinect, in which the coarse-to-fine iterative closest point (ICP) algorithm is used to map dense point clouds in order to estimate the pose of the Kinect camera. In their latest work, dynamic fusion [Newcombe et al., 2015] is achieved which is able to fuse a 3D model onto a slow-moving object.

2.2.3 Discussion

Both Kinect-like cameras and binocular-cameras have been widely-used in mobile robots and manipulation robots. For the binocular-cameras, the performance of depth sensing is determined by their stereo matching algorithms. Rigid object manipulation scenes usually are of discontinuous depth, and the depth changes drastically on the object boundaries. Therefore, segmentation-based stereo matchers are suitable for this task. While in clothes manipulations, the clothes surfaces are more likely to be continuous, hence correlation based local stereo-matching algorithms are likely to achieve a good performance. After a comprehensive literature investigation and experimental observations, the C3D matcher [Siebert and Urquhart, 1995] is used for depth sensing of the proposed clothes visual perception research.

2.3 Depth Data Analysis

Visual perception is one of the most important components of an autonomous system. In visually-guided manipulation tasks, 3D data (depth map or point cloud) is more robust than RGB data in terms of representing the shape, size, and landmarks of objects. In this thesis, one of the objectives is to advance the visual perception of garments and explore the effectiveness of visual perception in garments manipulation. Therefore, here a survey on 2.5D/3D surface analysis is provided.

2.3.1 Geometric Features

Curvatures are important low-level features for depth map analysis as the curvatures indicate the shape of 3D object. They can be directly used for depth based recognition work [Gordon, 1992] and generate higher-level features such as surface ridges and valleys [Belyaev and Anoshkina, 2005, Ohtake et al., 2004], shape index [Koenderink and van Doorn, 1992], etc.

Brady et al. [Brady et al., 1985] modelled the 'step' and 'roof' like edges on the depth surface with the corresponding discontinuities of the principle curvatures. In their approach, the surface edge is modelled as 'step' if its maximal curvature is changing rapidly in its direction, and as 'roof' if the maximal curvature is on the extra.

Ridge lines contain the essential topology information of a geometric surface. Ohtake et al. [Ohtake et al., 2004] proposed a ridge valley line detecting approach on mesh surface after an implicit surface fitting. Later, Belyaev et al. [Belyaev and Anoshkina, 2005] gave the geometrical definition of ridge points − the positive extrma of maximal curvature. Shape index [Koenderink and van Doorn, 1992] is a classical 2.5D surface shape analysis approach, which can classify 9 different shape types in terms of index integer values.

B-Spline surface [Rogers, 2001] is a classical 3D graphic modelling method, and can also be used to approximate implicit surfaces from observed surface points. In piecewise surface fitting, as opposed to typical least-square polynomial surface fitting, B-Spline surface fitting [Rogers, 2001] more easily achieves continuity between adjacent patches.

2.3.2 Depth Image/Point Cloud Registration

In early stage computer vision, feature registration/matching based methods are used to recognise the previously-seen objects. Moreover, this kind of method are also widely used to fuse multiple view depth map to obtain a solid 3D scene/object. In this procedure, depth map/point cloud registration techniques are used to estimate the rigid transforms.

Besl et al. [Besl and McKay, 1992] proposed an iterative closest point (ICP) algorithm to register a scene point cloud to a model point cloud. The algorithm starts with an initial registration. Then, in each iteration of their approach, the transform estimated from the latest registration is applied on a scene point cloud, the nearest point pairs are found using k-d trees [Friedman et al., 1977], and a new transform is estimated. This procedure is processed iteratively until the registration error is lower than a pre-defined threshold. Since the closest point pairs matching are used for the registration, this approach works provided the the rotation between the scene and model point clouds is reasonably small. It is worth noting that the approach [Besl and McKay, 1992] is able to handle the registration problem of other types of data, including line segment sets, implicit curves, parametric curves, triangle sets, implicit surfaces and parametric surfaces.

In the following research, various kinds of 2.5D/3D local descriptors [Johnson and Hebert, 1999, Lo and Siebert, 2009, Rusu et al., 2009, Steder et al., 2010, Tang et al., 2012] are proposed in order to acquire a better matching performance between two range images and point clouds. More details are introduced in Section 2.4.3.

2.3.3 Visual SLAM-Based 3D Reconstruction

Simultaneous Localization and Mapping (SLAM) has been developing for almost 30 years, in which visual dense-mapping SLAM approaches can be used for 3D reconstruction. These approaches can be grouped into monocular-SLAM and RGB-D SLAM, depending on the input data types. In terms of monocular-SLAM, the transform between current camera frame and key-frame is estimated, then a depth map is generated by triangulating and merged with the global map. But during the pose tracking, errors are accumulated, which is the most difficult problem SLAM needs to resolve. For example, in Davison's classic monocular SLAM [Davison, 2003], the visual features are used to estimate the camera frame transform, and camera states are modelled by 13 parameters, including 3D position, quanterian, motion, linear velocity and angle velocity; Extended Kalman Filter (EKF) [Welch and Bishop, 1995] is used to reduce the transformation noise. Newcombe [Newcombe, 2012] proposed a feature-based passive dense SLAM which follows a pipeline of frame-selection and depth-prediction, productive stereo and depth map fusion. LSD-SLAM [Engel et al., 2014] is the state-of-the-art dense-mapping (feature-less) monocular SLAM, in which the camera pose transform is estimated by minimizing the photometric errors among dense pixels and Gauss-Newton method is employed to solve this optimization problem.

SLAM-based 3D reconstruction approaches are able to offset the drawbacks appearing in single-shot stereo vision, e.g. occlusions, while their prerequisite is that the image frames have to be captured from distinctive views.

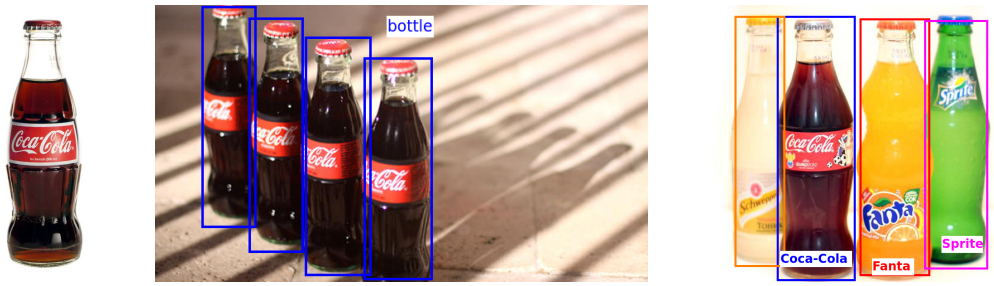
2.4 The Object Recognition Pipeline

Object recognition/identification is an extremely important problem in computer vision. Since clothes category recognition is one of the potential tasks this thesis aims to advance, the literatures about the general object recognition pipeline are reviewed here.

2.4.1 The Problems

For this field of research, there are three sub-problems researchers are working on: image classification, object detection and object recognition. To begin with, the difference and re-

lationships among these three problems are illustrated. In the image classification problem, the image for learning usually contains one major object[8], which is the learning object/target. For the object detection, the goal is to localise the interested object(s) from a wider scope. Sliding-window (exhaustive searching) [Dalal and Triggs, 2005] is a classic approach for object detection, in which the small detecting window slides through the image over different scales, and for each position and scale, the window is classified whether it has interesting object(s). In other words, each detection is an image classification problem where the detection window is treated as an image. As a consequence, the object can be detected (localised) as a rectangle bounding box in the image. The object detection problem can be binary-class or multi-class, and the category labels are likely to be 'coarse' labels (e.g. human, vehicle, etc.). Compared to object detection, the object recognition (identification) usually tends to recognize multi-class objects where the class information is 'finer' than that in object detection (e.g. children, adults, cars, vans, etc.).



(a) An image of a glass bottle. (b) The detection/localisation of glass bottles. (c) The recognition/identification of glass bottles.

Figure 2.7: The difference between image classification, object detection and object recognition.

Here is an example of the difference between these three problems. As shown in Fig. 2.7(a), in the image classification problem, the image itself is the learning object. In Fig. 2.7(b), the bottles are detected and localised as bounding boxes. While in the object recognition/identification problem, as shown in Fig. 2.7(c), the bottles can be identified by their brands. Additionally, for instance, in face image classification problem, the goal is to classify the image of which the majority is a face; in a face detection problem, the objective is to detect and localise all human faces in the testing image; in a face recognition problem, the goal is to identify whose face it is. In this thesis, the problem solved in Chapter 5 and Chapter 6 is recognizing/identifying the clothing categories (t-shirt, shirt, sweater, jeans, towel).

The differences among image classification, object detection and recognition are presented above. Now the relationships among them are illustrated. The essence of object detection and recognition is the image classification problem, because in the sliding window searching procedure, each detecting window performs an image classification task. However, in

[8]In other words, the objects are centred in the images, whilst the scene, texture images are excluded.

the following research, more advanced component-based object recognition or pixel-level semantic scene labelling are proposed. These are addressed essentially by advanced image classification methods. Section 2.4.3 introduces the image classification approaches from the early-stage to the state-of-the-art.

2.4.2 Searching/Localising Strategies

Figure 2.8: An example of selective searching. [Uijlings et al., 2013]

Sliding window has been extended to detect 3D objects in the point cloud [Song and Xiao, 2015] by sliding a 3D detecting cuboid in x, y, z directions. Instead of using exhaustive searching, Uijlings et. al proposed a selective search [Uijlings et al., 2013] to generate effective object localisation proposals that reduce the detecting space remarkably. In their method, image pixels within similar color space are first grouped into small regions, and then small regions are grouped hierarchically based on four types of similarity measurements: color, texture, size and content. During this hierarchical region grouping, localisation proposals interpreted as rectangular boxes are yielded from the regions. Besides, other widely-cited non-exhaustive searching strategies are: jumping window [Chum and Zisserman, 2007] and hough-transform-based searching [Maji and Malik, 2009].

2.4.3 Image Classification

Global Feature-Based Approach

Global features have been widely-used in the early research of image representation. These are: color histograms [Chapelle et al., 1999], boundary descriptors [Granlund, 1972, Kuhl

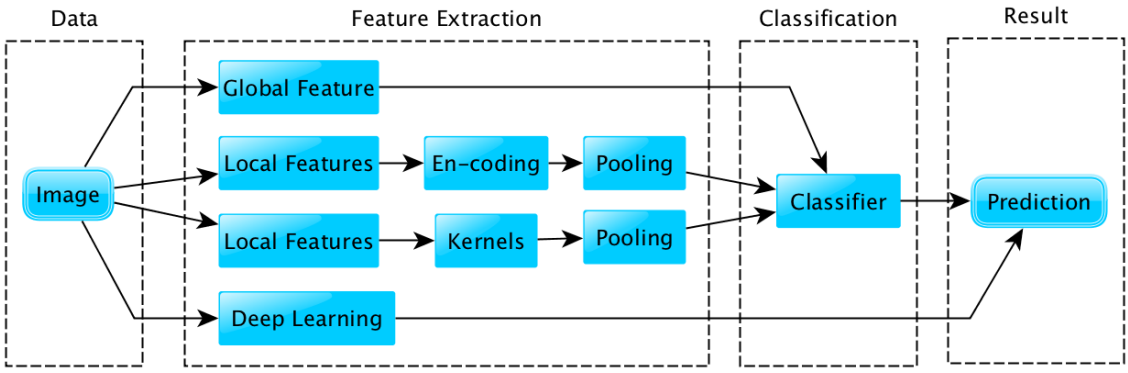


Figure 2.9: The image classification pipeline.

and Giardina, 1982], region descriptors [Hu, 1962, Khotanzad and Hong, 1990] and texture descriptors [Haralick et al., 1973, Ojala et al., 2002b].

Histograms, quantifying the statistical color information for all the image pixels, are widely used as global image descriptors in early-stage vision. Before calculating the histogram, the original RGB color information can be translated in different color spaces such as Grey, HSV, in order to improve the robustness [Chapelle et al., 1999].

The elliptic Fourier feature [Kuhl and Giardina, 1982] is the most widely-cited global feature for describing a contour, in which the Fourier approximation can be tuned from coarse to fine by increasing the number of coefficients. It has been adapted to contour-based image classification [Kuhl and Giardina, 1982] and hand-written characters recognition [Granlund, 1972]. Afterwards, shape context [Belongie et al., 2002] was proposed to match the 2D shapes and adapted to prototype-based object recognition due to its robustness on deformable shapes.

Moments describe the layout of a solid shape which is used to describe image regions. Image moments are centralised in order to achieve translation invariance [Hu, 1962]. Afterwards, rotation-invariant moments were proposed [Khotanzad and Hong, 1990] by using polar representation.

Local Binary Patterns (LBP) descriptor is generated by comparing the values between a central pixel and its neighbours (usually 3×3 regions). These patterns can be counted into a histogram to obtain a global representation. Multi-scale LBP descriptor achieve outstanding performance on texture classification for small-scale datasets [Ojala et al., 2002b].

Local Feature-based Approach

The most widely-cited 2D image local descriptors are SIFT [Lowe, 1999b, 2004] and HoG [Dalal and Triggs, 2005, Lowe, 1999a]. Both of them are histograms of image gradient

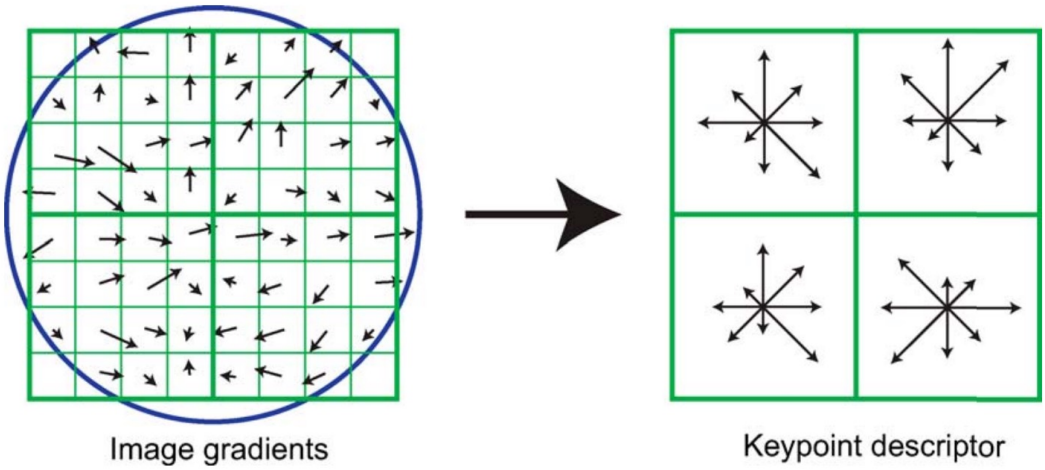
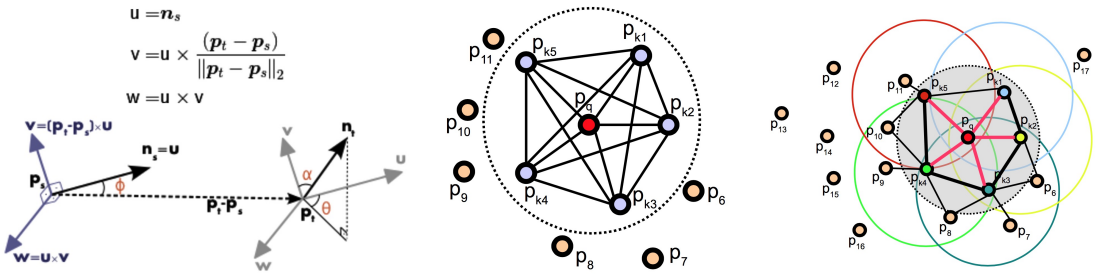


Figure 2.10: Example of SIFT descriptor. In the left figure, the gradient's orientation and magnitude are calculated in a set (here is 8×8) of cells. After Gaussian weighting, the SIFT descriptor is formed as sub-descriptors from 2×2 grids and each of these is histogram of gradient histogram accumulated by the magnitudes of samples within the subregion (as shown in the right figure).

orientations. To be more specific, the local image patch is divided into grids, then grids into cells (blocks), then histograms of gradient orientations are calculated in each cell (block) and finally grouped together. A more detailed example is given in Fig. 2.10. Histograms of gradient orientations are originally translation-invariant; moreover, SIFT features achieve scale-invariance by extracting the key points on the extrema of DoG (Difference of Gaussian) pyramid and achieve rotation-invariance through setting the canonical orientation. It is worth noting that in HoG features, the orientations are weighted by their gradient magnitudes, thereby enhancing the outline curves of the object. In the following research, more efficient or effective local descriptors are proposed: SURF [Bay et al., 2006], BRIEF [Calonder et al., 2010], BRISK [Leutenegger et al., 2011], FREAK [Alahi et al., 2012], etc.



(a) The α, β, θ angles.

(b) The Point Feature Histogram (PFH).

(c) The Fast Point Feature Histogram (FPFH).

Figure 2.11: The Point Feature Histogram (PFH) and Fast Feature Histogram (FPFH) features. The red point is the query point. In PFH, the points are fully connected in the sphere, while in FPFH, points are only connected with the query points, and features in neighbouring spheres are grouped together.

For 3D object classification[9], the following 3D local descriptors can also be used to describe the local components of 3D objects.

Johnson et al. proposed a rotation invariant feature Spin Image [Johnson and Hebert, 1999] for point cloud or 3D mesh registration and 3D object recognition. The Spin Image is extracted on an original point, and every 3D point within a specified distance is projected into τ, υ coordinates, where τ and υ values indicate the projected distance between the point and origin on the tangent plane and the surface normal. The τ and υ values are accumulated into uniform bins to construct the 'Spin Image'.

Point Feature Histogram (PFH) [Rusu, 2010], proposed by Rusu in 2009, is on the basis of robust surface normal estimation via PCA and fast point cloud localisation by tree structures [Rusu and Cousins, 2011]. As shown in Fig. 2.11(b), for each query point, its neighbours located in a local sphere space are retrieved; then for each unrepeated pair of the points within this sphere, the α, β, θ angular values shown in Fig 2.11(a) can be calculated from the estimated surface normals. The final descriptor can be formed by quantifying these angular values into three-dimensional histograms. Later on, PFH was accelerated by only connecting query points with neighbours, as opposed to connecting every pair in the sphere. This yields Fast Point Feature Histogram (FPFH) [Rusu et al., 2009], shown in Fig. 2.11(c).

Besides, Lo et al. [Lo and Siebert, 2009] extended the 2D SIFT [Lowe, 2004] descriptor to a depth map. Steder et al. [Steder et al., 2010] proposed a sparse local feature NARF (Normal Aligned Radial Feature) extracted on a depth map. NARF usually appears on the edges of the object where the shape of the object's surface is unstable. Tang et al. [Tang et al., 2012] proposed the HONV (Histogram of Oriented Normal Vectors) descriptor by accumulating the surface normal vectors in sphere coordinates. In [Rusu, 2013], Shape Index [Koenderink and van Doorn, 1992] has been extended to analyse the surface shape type of a point cloud on a mesh surface.

For 3D features extracted from a 3D point cloud, searching the nearest neighbour or points within a radius is much more expensive than it is in a depth map, although tree-structure-based searches reduce the complexity remarkably. The feature descriptors formed from the histogram only retain statistical information of a 3D surface, this process is irreversible. Moreover, from the practical experience and literature investigation, these reviewed features devised for describing 3D rigid objects are unlikely to be distinctive when they are applied to deformable surfaces such as clothes.

[9]Here, 'object' is preferably used instead of 'image' because the 3D data has spatial differences on both x-y and z (depth) directions.

Encoding

After local descriptors are extracted from the image, an en-coding (coding) method is required to transform the local descriptors into a vocabulary-space to acquire higher-level representation. One of the most classic coding methods is bag-of-features (BoF)[10]. In BoF-Like methods, the local feature descriptors are treated as 'visual words' and unsupervised learning techniques such as k-means, GMM and mean-shift are employed to train the 'codebook' (or dictionary). In the coding procedure, each 'word' (local descriptor) in the image is matched with its nearest codebook entry, and the accumulation of each codebook pattern is used as global representation (this process is also called sum-pooling [Lazebnik et al., 2006]). Vector Quantization (VQ) [Deng and Manjunath, 2001] adapts the same coding process, but uses the vectorized raw image patch or filtered image patch instead of local descriptors. For this characteristic, VQ is more popular in texture recognition.

Sparse coding methods advance the performance of classical BoF methods. The goal of sparse coding is to reconstruct the local descriptor from a weighted linear combination of codebook bases. These weights (which are called sparse code) have to be sparse in order to be distinctive. For this purpose, the loss functions of sparse coding methods usually have two terms: the reconstruction-error term and the sparse-penalty term. More specifically, the reconstruction-error term captures the error between local descriptors and the reconstructions from weighted linear combination; the sparse-penalty term enforces the code to be sparse. In Lee et al. [Lee et al., 2006], L2 norm reconstruction-error and L1 norm penalty terms are used. In order to minimise this loss function, the L2 and L1 terms have to be optimised iteratively. Although both of these are convex problems, the complexity of optimisation is not satisfactory. In subsequent research, a more effective and efficient Locality-constrained-Linear Coding (LLC) [Wang et al., 2010] is proposed, in which the locality (local coordinates coding constraint) is utilised to enforce sparseness. In the coding procedure, LLC first retrieves a small number of codebook base using nearest neighbours and then applies linear coding in this local coordinate. The performance of LLC on bench-mark image classification dataset demonstrates that the effectiveness and efficiency can coexist.

Kernel Methods

Fisher Vector Kernel is based on Fisher Discriminant Analysis [Jaakkola et al., 1999] and was adapted to the classification problem. The intuition of Fisher kernel is to project examples of different classes to a separable 1D space. For the new projected distributions, the goal of learning is to maximise the distance between the means of different classes and minimising their respective variances. Thus, the projected examples of different classes are distributed

[10] In different literature, it is also called bag-of-words or bag-of-visual-features.

as far as possible and the overlapping of the said distributions is as small as possible.

In subsequent research [Perronnin and Dance, 2007], Fisher Vector Kernel is adapted to image classification. In this work, GMM is applied to train the codebook from local descriptors, where the probabilities of local descriptors for each codebook base can be generated. Thereafter, first order and second order statistics of Fisher Vector Kernel are used for encoding. Fisher Vector Kernel-based approaches have been adapted to image retrieval [Perronnin et al., 2010] and face recognition [Simonyan et al., 2013]. Furthermore, a simplified version of Fisher Vector Kernel - VLAD (Vector of Locally Aggregated Descriptors) coding [Arandjelović and Zisserman, 2013, Jégou et al., 2010] has been devised for image retrieval tasks. It is worth noting that the Fisher Vector Kernels can be adapted into any discriminative classifiers to complete the classification.

Fisher Vector Kernel-based encoding methods are the state-of-the-art for image classification using hand-crafted features, whereas their drawbacks are also obvious - the computation of kernel is very expensive. Although Perronnin et al [Perronnin et al., 2010] made a substantial improvement on large-scale data, the memory consumption and time consumption are still expensive compared to sparse-coding based methods.

Pooling

In order to summarize the local representation into an image-level global representation (codes), pooling is required. Sum-pooling [Lazebnik et al., 2006] and max-pooling [Yang et al., 2009] are the two widely-used pooling methods. Sum-pooling sums the corresponding values (value of the code) among all the codes with respect to each codebook base, while max-pooling uses the largest value among all the codes with respect to each codebook base as the global representation. It is worth noting that the coding process can be conducted in a spatial pyramid in order to emphasize the spatial information [Lazebnik et al., 2006, Yang et al., 2009], and by doing this, the performance can be improved provided the images have a strong spatial correspondence.

Classification

The final step of the image classification pipeline is "classification" – identifying to which of a set of categories an unknown example belongs. In this section, the state-of-the-art discriminant classification approaches are reviewed. The goal of discriminative classification is to fit the classifier model to training data and the learnt model should able to predict unknown testing data.

In the early literature, k-Nearest-Neighbour (kNN) [Cover and Hart, 1967] and decision tree [Safavian and Landgrebe, 1991] are the most widely-cited discriminant algorithms. kNN is a

classic non-parametric, lazy classifier, in which a simple majority voting strategy is adapted to predict the unknown examples. With this simple mechanism, it achieves satisfactory performance on a range of pattern recognition problems. Decision tree [Quinlan, 1986, Safavian and Landgrebe, 1991] is a hierarchical tree structure classifier in which each tree node splits the examples into two sub-sets through thresholding the values in a specific dimension, and the dimension for splitting is chosen by calculating the information gain.

At that stage, researchers found that the classification performance can be remarkably improved by combining weak-classifiers. For this motivation, various ensemble methods were proposed: bagging [Gregorski et al., 1994], Random Forest [Ho, 1998], AdaBoosting [Rätsch et al., 2001].

In the bagging algorithm [Gregorski et al., 1994], subsets of examples are sampled randomly from the whole training set to get the relatively independent weak classifiers. Voting is then used to obtain the final prediction from the predictions of weak classifiers. Random Forest [Ho, 1998] is essentially a collection of decision trees. For the training procedure, Random Forest has two 'random' processes: the first is, for training each decision tree, a sub-set is selected randomly from the training examples; the second is, only part of the data dimensions are chosen randomly for training. AdaBoosting [Rätsch et al., 2001] is a more complex method that is able to reinforce the training on incorrectly classified examples. Every example is initialized with a uniform sampling distribution; a certain percentage of training examples are sampled for training and the rest is for evaluating. After each iteration of training, the incorrectly classified examples in the evaluation set are assigned higher possibilities to be sampled, and thus they have more chance to be sampled for training in next iteration. And the trained classifiers are weighted according to the performance of evaluations.

Support Vector Machine (SVM) is almost the widely-cited discriminative classifier, benefiting from max-margin, soft-boundary mechanism and kernel function. In the training procedure, the objective function is to maximizing the margin of examples from positive and negative classes, while the decision boundary can be 'soft-boundary' with tolerance for misclassified examples. SVM also supports various kernels, and compared to linear kernels, non-linear kernels usually are able to enhance the classification performance but increase the complexity of computation.

Gaussian Process (GP) [Rasmussen, 2006] is a non-parametric model for regression and classification problems. In GP regression problems, the conditional probability of latent variables w.r.t testing examples given training examples, testing examples, latent variables w.r.t training examples is modelled as a multiple-variant Gaussian distribution. For the classification problems, the key problem is to estimate the posterior of latent variables given training data. Several inference methods such as Laplace Approximation [Rasmussen, 2006], Expectation Propagation [Rasmussen, 2006], are proposed to estimate this distribution. GP is also

kernel-based method since the covariance matrices are calculated by kernels that determine the estimation of distribution.

To the best of the author's knowledge and practical experiences, it can be deduced that, for strong discriminative classifiers such as SVM and GP, the performance is unlikely to be substantially improved if they are embedded into ensemble methods. On the other hand, the feature combination or non-linear feature fusion is likely to produce a substantial improvement.

Deep Learning

The most widely-used layers of CNN architecture [Jarrett et al., 2009, Krizhevsky et al., 2012, Simonyan and Zisserman, 2014] can be categorised as follows: *Convolution Layer*, applying locally-connected neural network weights to input signal via convolution; *Rectification Layer*, simply applying non-linear transform to input signal (to convert negative values to zeros or absolute values) in order to reduce the training complexity; *Local Contrast Normalization Layer*, normalizing the local contrast by subtracting the Gaussian weight mean; and *Pooling and Sub-Sampling Layer*, generalizing and summarising the signals by calculating the mean or average over neighbours, in order to enhance the spatial robustness.

CNN achieves great success on a range of image classification problems. In Krizhevsky et al. [Krizhevsky et al., 2012], CNN was applied to the ImageNet challenge, which was able to classify images of more than 1000 categories trained from 1.2 million images. Ciresan et. al [Ciresan et al., 2012] proposed a multiple column deep CNN where the corresponding values of CNNs are averaged with respect to each class before fitting into soft-max function. Their reported results achieve a substantial improvement on hand-craft character recognition problems.

In [Donahue et al., 2013, Jia et al., 2014], pre-trained CNN is used to produce visual features for the general object classification problem. More specifically, these features are yielded by forward-propagating images to the second last layer (before soft-max layer), since at that layer features are likely to be linearly separable. In subsequent work [Girshick et al., 2014], CNN together with selective searching [Uijlings et al., 2013] are applied to object detection.

CNN has recently been applied to the object detection/localisation problem. After CNN feature was adapted to region-based object detection [Girshick et al., 2014], Girshick proposed a fast Region-based CNN (R-CNN) [Girshick, 2015] which can predict the region proposals for the potential objects to be detected. This regression network starts with convolution layers, and the object region's (Region of Interest) features go through fully-connected layers to classify the object categories and regress localisations. As a consequence, given an unknown image, the region proposals of objects with respect to different categories can be

predicted. Subsequent work [Girshick, 2015, Ren et al., 2015] accelerate this process by merging classification CNN with region-based CNN.

Moreover, it is worth noting that CNN can also be adapted to dense (pixel-wise) scene understanding. In Farabet et al.'s approach [Farabet et al., 2013], CNN feature is used to label the scene where conditional random field (CRF) is used to model the joint probabilities between neighbouring pixels in order to refine the prediction map. Long et al. [Long et al., 2014] directly applied the CNN trained from patches to images of arbitrary sizes to get the semantic prediction map.

The key difficulty of training CNN is to set the network architecture. In order to adjust the neural network architecture structure and tune parameters of each layer, Zeiler et al. proposed a deconvolutional neural network [Zeiler et al., 2011] which is able to reconstruct the input in multiple layers by learning sparse code of a specific image (or its feature maps) from learnt filters common to all images. In their subsequent work [Zeiler and Fergus, 2014], a better visualization for features of a trained CNN is provided. This visualization illustrates how the network works and how to design the neural network architecture.

The performance reported in [Russakovsky et al., 2014] shows that the deep learning based methods have advanced the performance of hand-crafted features (global and local features) based methods on large-scale bench-mark image classification datasets. The prerequisite of deep learning is that the training requires large amounts of data. Take CIFAR[11] and ImageNet[12] for example: the former has 80 million images for 100 categories and the latter has 14 million images for more than 30,000 categories. For RGB images, powerful image search engine (e.g. Google, Bing) facilitates the data collection of large-scale datasets. On the other hand, for the specific object recognition for manipulation tasks, where the training data is collected manually in the lab, the collection of such a large-scale datasets becomes extremely difficult. For clothes recognition dataset[13] used during the production of this thesis, there are only 2100 RGBD images for 5 categories of clothes.

2.4.4 Other Methods for Object Recognition

Apart from the image classification based methods investigated in the former sections, there exist other types of approaches to recognise 3D objects. In Rusu et al. [Rusu et al., 2008], the geometric shape e.g. edges, boundaries, planes, of a point cloud can be identified by mapping the real point cloud data with geometric model. In Rusu [Rusu, 2009], a more comprehensive research in terms of a 3D household scenario is reported including geometric surface analysis, 3D visual feature based object recognition, and semantic point cloud

[11] available at: $http://groups.csail.mit.edu/vision/TinyImages/$
[12] available at: $http://www.image-net.org$
[13] available at: $https://sites.google.com/site/clopemaclothesdataset/$

labelling. In Song et al [Song et al., 2015], a SIFT-based matching and pose estimation method is adapted to recognising all the objects of a pre-trained, known, indoor environment. The color and surface normals are used as the prior to boost the semantic labelling performance of obstructed or low-confidence regions.

2.4.5 Discussion

The registration-based methods introduced in Section 2.4.4 are more likely to register rigid objects. Some local descriptors are of a greater degree of invariance to deformable shapes, and as a consequence, the registration-based methods are adapted to lightly-deformable objects such as toys, faces, etc. Whereas, for highly-deformable clothing registration, when positions and shapes of landmarks change, the registration is very likely to fail.

The deep learning based methods are the state-of-the-art approach for large-scale image classification problems, but as it is investigated in Section 2.4.3, massive training data is required for these methods, which cannot be satisfied in this thesis. Training a deep neural network requires to optimise 100k - millions of parameters (neural network weights), training from insufficient training data is very likely to lead a over-fitting. Using pre-trained network can reduce the required amount of training data and eliminate over-fitting. However, the prerequisite is that the training data between pre-training and fine-tuning should be highly correlated. For instance, the network pre-trained from RGB data is unlikely to be able to work on 3D data, or the network pre-trained from 3D rigid objects is unlikely to be able to be adapted to classifying different categories of clothing. For the problems intended to be resolved in this thesis, it is difficult to find a proper public large-scale dataset for pre-training.

For hand-crafted features, in general, local features+encoding+pooling-based descriptions achieve better representation than global description but lead to a higher computation complexity. Hence, this is a trade-off between performance and efficiency. In this thesis, both global features and local features are further investigated.

2.5 Manipulation of Rigid Objects

This section investigates the state-of-the-art visually-guided manipulation approaches for grasping rigid objects.

Saxena et al. [Saxena et al., 2008a] proposed a supervised learning based approach to detect the grasping position on rigid objects from RGB images, in which texture filters are used for visual features and a logistic regression based classifier can be trained on the pixel level. Later, they complete the grasping pose estimation [Saxena et al., 2009] and combine the vi-

sual perception with the kinematic description [Saxena et al., 2008b] to specify the positions of figures of different types of robot hands.

Lenz et al. [Lenz et al., 2015] devised two-layer-deep neural networks (CNN) to explore the grasping position and poses from RGBD data. More specifically, they annotated the grasping position using a rotatable rectangle on the image frame, from which the RGB data and surface normals are extracted and fed into the networks. The first neural network is employed to reduce the exploration space and a deeper neural network is used on the preliminary search results to the exhaustively explore all the possible poses. CNN has the advantage of straight forward encoding from raw data to prediction. In their work, the surface normals are extracted instead of using raw data, and as a consequence, the learnt network becomes an encoder between visual features to prediction result.

Kopicki [Kopicki et al., 2015] further investigated the grasping of unknown rigid objects with different hand configurations using a humanoid robot hand. In his research, two probability density functions (PDF) are modelled via Silverman's kernel density estimation method [Silverman, 1986] in order to obtain the density distributions of grasping location and configuration type, respectively.

Besides rigid object grasping, Goldberg et al. [Goldberg et al., 2005] proposed a method to grasp the deformable parts of an object. In their work, the topological shape of an object is modelled by polygons with finite degrees of freedom, and thereby the stiffness of the object's components can be measured.

The rigid objects (usually kitchen objects e.g. bottles, bowels, plates, cooking tools, etc.) are relatively smaller than garments, and therefore the grasping detection task can be addressed by exploring different grasping poses centred by the objects. However, these methods cannot be directly adapted to deformable garment grasping. The difficulty is that, more delicate configuration parsing and dexterous grasping are required in order to fetch small landmarks on garments (e.g. wrinkles, collars, cuffs, etc.).

2.6 The State-of-the-Art of Clothes Perception and Manipulation

This section reviews the literature of clothes perception and manipulation. A historical perceptive is firstly presented in Section 2.6.1, showing the statistics of literatures during past 20 years. Subsequently the early-stage research in this area is introduced in Section 2.6.2. The state-of-the-art of clothes perception and manipulation is categorised and introduced in three parts: visually-guided clothes manipulation (Section 2.6.3), clothes recognition (Section 2.6.4) and interactive perception (Section 2.6.5). The discussion is given in Section 2.7

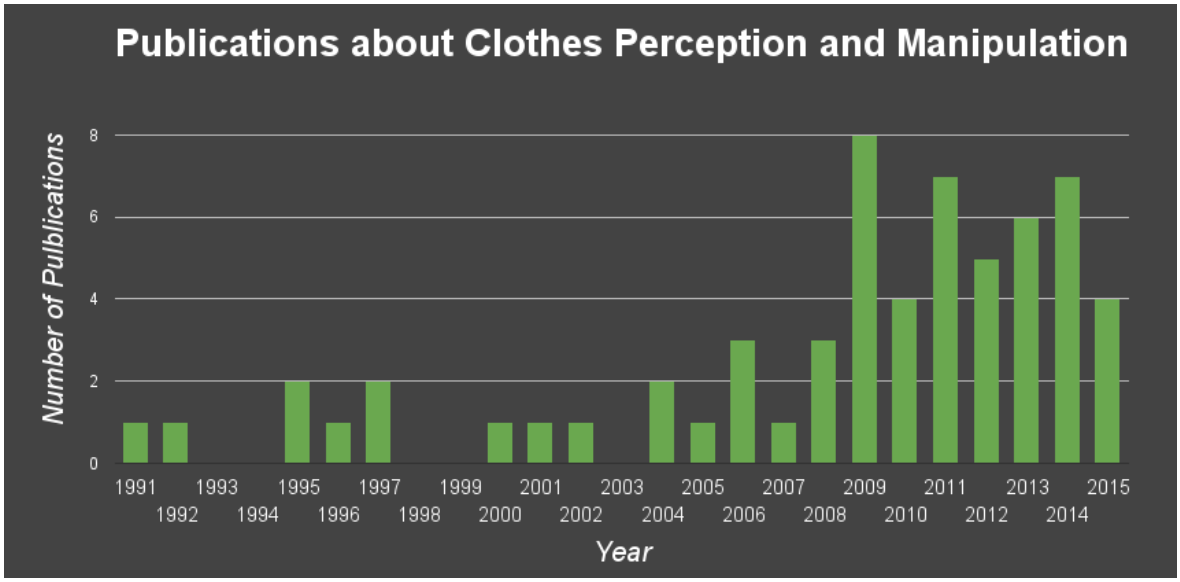


Figure 2.12: Publication statistics in robot clothes manipulation research.

which details the limitations of the state-of-the-art and their potential solutions.

2.6.1 From Historical Perspective

Robot clothes manipulation is a novel research topic, and much fewer research work have been published as compared to robot manipulation of rigid objects. With the investigation of the history of clothes manipulation, the statistics of publications in this area are shown in Fig. 2.12. According to the bast knowledge investigated in this thesis, the first publication on clothes manipulation was published in 1991, and in the following 16 years, this area has been quiet, with less than 3 papers published every year. In 2008, UC Berkeley [Maitin-Shepard et al., 2010] achieved the first autonomous robotic laundering pipeline to grasp, unfold and fold towels using the PR2 robot, which was a critical moment in the history. Subsequently, research in perception and manipulation has developed rapidly, and researchers are working on each subtask of an autonomous laundering pipeline [Willimon et al., 2011a]: grasping clothes from a pile [Ramisa et al., 2012], recognising the clothing categories [Kita et al., 2009a,b, Li et al., 2014a,b, Willimon et al., 2013a, 2011b, **?**], unfolding the garments [Cusumano-Towner et al., 2011, Doumanoglou et al., 2014a,b, Willimon et al., 2011a], pose estimation [Kita et al., 2004, 2009a,b, Li et al., 2014a,b] and garment folding [Maitin-Shepard et al., 2010, Miller et al., 2012, Stria et al., 2014b, Van Den Berg et al., 2011].

2.6.2 Early-Stage Research

Howard et al. proposed a method to estimate the deformation characteristics of deformable objects from manipulating them [Howard and Bekey, 1997], and they verified their method in physical simulations.

Hamajima et al. [Hamajima and Kakikura, 2000] developed an early-stage robotic laundering system including grasping, unfolding and folding. They proposed an unfolding approach by grasping the hemline of the hanging garment. Limited visual knowledge such as edge detecting, Gaussian filtering and thresholding are used to locate the grasping positions and detect the hemlines.

Yamakazi et al. [Yamazaki and Inaba, 2009] proposed applying Gabor filtering to an intensity image for wrinkle detection. Intensity image based features are extremely sensitive to the intrinsic texture observed on the manipulated cloth. This results in reduced ability to distinguish concave and/or convex regions around wrinkled areas.

From the above investigation, it can be found in early-stage research that, the research focused on estimating the physical attributes of deformable materials and designing manipulation strategies. Constrained by the development of computers and depth sensors at that time, the visual perception was very preliminary (basically are image precessing techniques such as filtering, thresholding).

2.6.3 Visually-Guided Clothes Manipulation

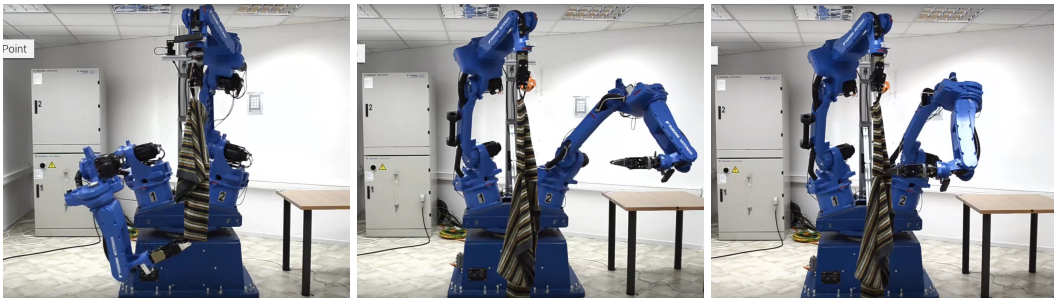
This section investigates visual perception approaches that guide the robot to conduct clothes manipulation tasks. These tasks include clothes grasping, category recognition, unfolding, flattening and folding.

Clothes Grasping

For garment grasping, Ramisa et al. [Ramisa et al., 2012, **?**] proposed a grasping position detection approach using RGBD data. In their approach, SIFT and GDH (Geodesic Distance Histogram) local features are extracted on wrinkled regions in the RGB and depth domain, respectively. After Bag-of-Features coding, two layers of SVM classifiers are trained with linear and χ^2 kernels. During the testing phase, a sliding window method is employed to detect graspable positions. After detection, 'wrinkledness' measurements calculated from the surface normals are used to select the best grasping position. The main drawback of these approaches is that SVM-like sparse classifiers are not able to quantify the quality of grasping predictions. Moreover, learning from human-labelled data is an interpretation of human knowledge, which potentially might not suit a robotic testbed, e.g. there are constraints of

the offset of field of view between human and robot, dissimilarities between humans' and robot's end effectors, and so forth.

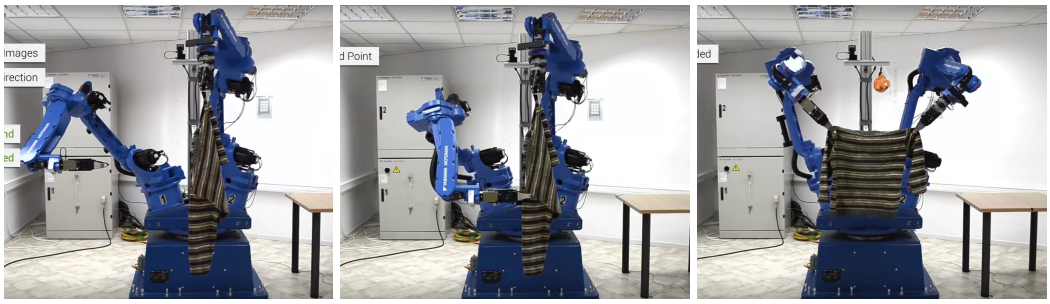
Clothes Unfolding



(a) Hanging the clothes and grasping the lowest point.

(b) Detecting the unfolding point (here it is the shoulder).

(c) Grasping the detected unfolding point.



(d) Hanging and detecting the other unfolding point (the other shoulder).

(e) Grasping the detected unfolding point.

(f) Unfolding is completed by grasping two unfolding points.

Figure 2.13: The procedures of robotic clothes unfolding [Doumanoglou et al., 2014a].

Most of the unfolding work is to unfold the garment in hanging poses. For garment unfolding, the key step is to detect the grasping positions for unfolding (e.g. corners of a towel, shoulders of a shirt, waist of a pant, etc.). For instance, Cusumano et al. [Cusumano-Towner et al., 2011] proposed a multi-view based detection approach to find the two corners of the same edge for unfolding towels. Afterwards, Doumanoglou et al. [Doumanoglou et al., 2014a,b] adapted unfolding for all categories of clothes, in which active random forests and hough forests are used to detect the grasping positions. More details of Doumanoglou et al.'s unfolding method is demonstrated in Fig. 2.13. Subsequently, Li et al. proposed an interactive unfolding strategy [Li et al., 2015b], in which the importance of grasping positions for unfolding is modelled by Gaussian density function, and in each iteration, garment's pose is estimated [Li et al., 2014a] and garment is re-grasped towards the more important position. Willimon et al. [Willimon et al., 2011a] propose an interactive perception-based strategy to unfold a towel on the table. Their approach relies on detecting depth discontinuities on corners of towels. For each iteration, the highest depth corner on the towel is grasped and

pulled away from its centre of mass. This approach is constrained to a specific shape of cloth (square towel), hence it is unlikely to be extended to other clothing shapes.

Clothes Folding

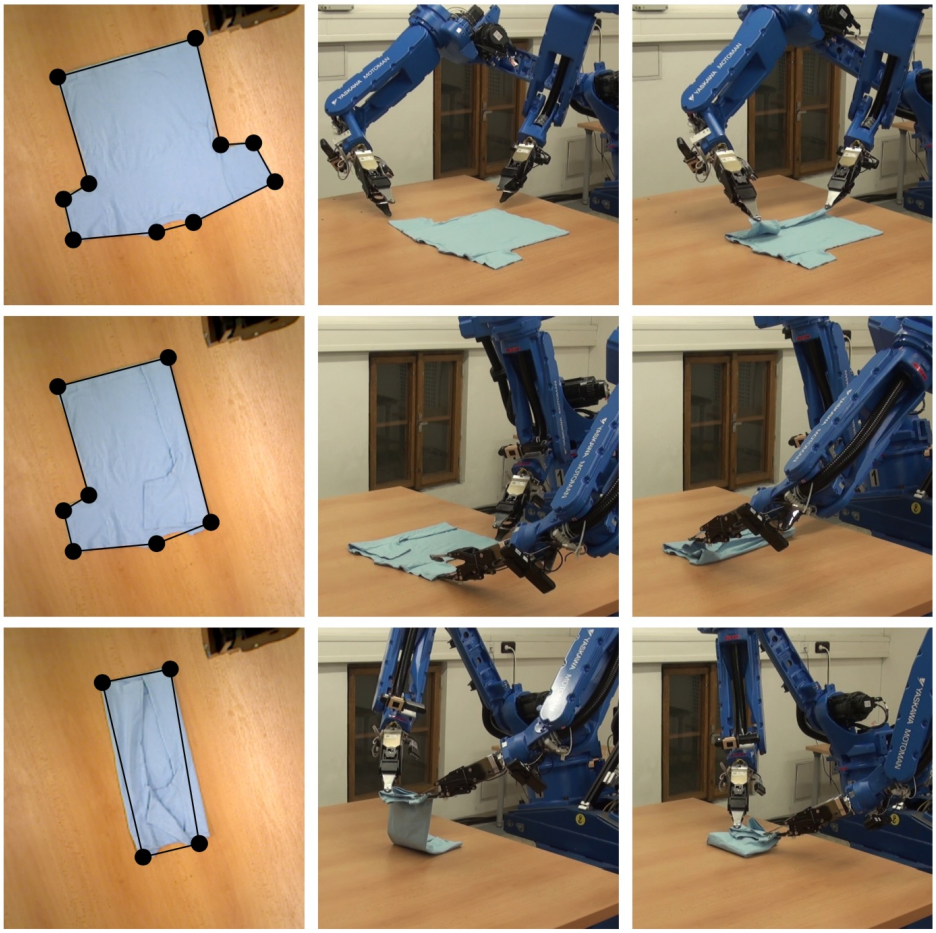


Figure 2.14: The procedure for folding a t-shirt [Stria et al., 2014b]. The clothes folding approaches usually follow a heuristic folding strategy for each type of garment. For example, folding a t-shirt starts with folding two sleeves and then folding the square body twice.

The final step for the laundry pipeline is folding. In that respect, Miller et al. [Miller et al., 2011] modelled each category of clothes with a parametric polygonal model. They proposed an optimisation approach to approximate a polygonal model to the contours of clothing obtained by segmentation. Thereafter, this approach was used to fold garments based on a gravity-based folding [Van Den Berg et al., 2011] and geometry-based folding [Miller et al., 2012] strategies. Stria et al. [Stria et al., 2014b] proposed a more efficient polygonal model which was able to accelerate matching by detecting landmark points. The robot demonstration of Stria et al.'s approach is shown in Fig. 2.14. The parametric polygonal models are unlikely to be adapted to other laundering tasks. This is due to the fact that polygonal models cannot recognise or track the state of a garment during folding, i.e. the state of the clothes are

discretized into several steps during folding and the transition between states are assumed to be known. Likewise, only 2D contour of garment segmentation is considered, and therefore, visual knowledge is insufficient for other laundering tasks that need to perceive the garment surface.

Apart from perception, Li et al. [Li et al., 2015b] optimise the folding trajectory from simulated folding. In their method, the parameters of friction and bending of the clothes for simulation are pre-tuned to approximate the reality of the table and garment. The objective function is set as the offset between optimal folding position and simulated folding position with a penalization of the length of trajectory.

2.6.4 Clothes Recognition

Recognizing From Hanging Configurations

Existing clothing category recognition systems can be divided into two groups: those based on perceiving the state of clothing as observed from RGBD data [Willimon et al., 2013a, 2011b] and those based on physical simulated models [Kita et al., 2004, 2009a,b, Li et al., 2014a,b].

Approaches based on the former include Willimon et al. [Willimon et al., 2011b] who proposed an interactive pipeline to recognise clothing categories from hanging poses by acquiring multiple views of each item while a robot-arm rotates the item. Visual features used for recognition are based on binary silhouettes and clothing edges. Nearest neighbour classification in feature space is used for categorisation.

Approaches based on simulated models include Li et. al [Li et al., 2014a] where they proposed a supervised learning approach for recognising clothing categories in hanging poses. They used dense SIFT and sparse coding [Lee et al., 2006] as the underlying visual representation. They then optimised their representation by using a binary volumetric representation to achieve real-time performance [Li et al., 2014b]. Physical simulated models benefit from the ability to generate the necessary training data for training a classification model. However, clothing items of free configurations are extremely difficult to simulate, e.g. [Wong, 2014] reports the difficulties in simulating: (a) particle inter-collisions between clothing surfaces; and (b) the static and dynamic frictions between garments and a supporting surface.

Physical simulated studies [Kita et al., 2009a, Li et al., 2014a,b] mainly report recognition of garments in hanging configurations, since the configuration space is greatly reduced. However, a large robot with a large working space is required. Hence, medium sized robots cannot be used to manipulate adult garments. As free-configuration clothing typically presents several occlusions, and a much larger configuration space, the performance of existing simulated

approaches within these scenarios is limited. Compared to the widely-used silhouette, RGB patterns and component features, 3D clothing configurations provide, as presented in this thesis, generic and invariant material attributes which are robust to occlusions and random pose configurations. As describe in the following sections, this thesis therefore proposes a pipeline incorporating an enriched visual description of clothing material type attributes.

Recognizing From Free Configurations

Thereafter, Willimon et al. [Willimon et al., 2013a] proposed a mid-level representation, in which 17 mid-level semantic classifiers of clothes attributes (e.g. collar, buttons, hem, colours, patterns, etc.) are trained from low-level RGBD features. SIFT [Lowe, 2004], Fast Point Feature Histogram (FPFH) [Rusu et al., 2009] and Bag of Features (BoF) coding [Csurka et al., 2004] combined with global features are used as the low-level representation while the output of the 17 mid-level semantic classifiers encode a binary representation of the garment. In this thesis, however, the proposed clothes recognition approach focuses on material types and generic 2.5D surfaces. The specific semantic components are not considered since they are susceptible to occlusion. Also the RGB information is ignored since it is not a stable representation for clothing categories. RGB data requires a sheer number of training examples to obtain classification results above chance.

Similarly, Ramisa et al. [Ramisa et al., 2013] devised a 2.5D local descriptor, FINDDD by constructing surface normal histograms over quasi-equidistant bins in the Euclidean space. They used FINDDD together with Bag of Features (BoF) for clothes category recognition in free-configurations. The approach proposed in this thesis, however, differs on the robustness of intra-class dissimilarity. For example, they used one polo-shirt to represent the category of polo-shirts. For clothes category recognition using depth data, the extra-class similarities are much larger than intra-class similarities; however, there is no guarantee that intra-class similarities can be neglected. Likewise, their experimental results consist of recognising unknown configurations of known clothes. Dividing training and testing sets on configuration-level instead of clothing-level opens up the possibility of over-fitting.

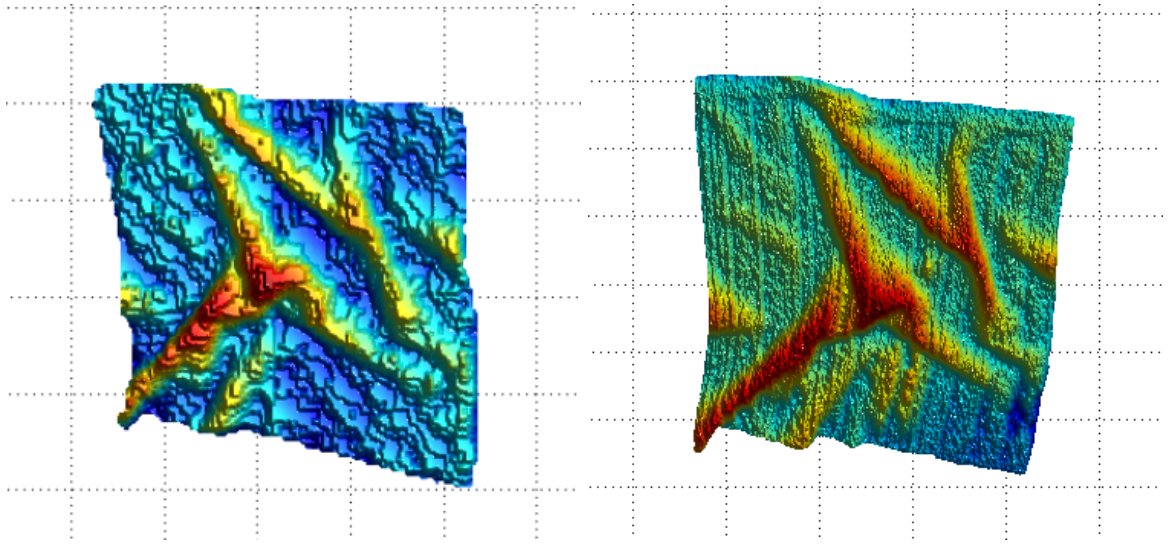
Pose estimation

From clothes category classification, researchers have attempted to infer the garment's pose. Approaches have focused on hanging clothes, for instance, Kita et al. [Kita and Kita, 2002] who first proposed a naive physical model for garment pose estimation. Their approach consists of matching 2D silhouettes to different simulated poses. A 3D point cloud of the observed garment is locally aligned to a simulated model, improving the robustness of the non-linear registration between real and simulated garments [Kita et al., 2009a]. Further-

more, the simulated model is used to estimate and optimise the hand pose for dual-arm manipulation [Kita et al., 2009b]. Li et al. [Li et al., 2014a] classify clothes types and poses using synthetic training data. In their approach, dense SIFT and sparse coding techniques are used as visual representations and SVM based techniques are used for the classification step. Thereafter, they propose [Li et al., 2014b] a volumetric binary description to improve their algorithm to execute in real-time. Among the pose estimation work of hanging garments, raw point cloud or engineering features are used to match or classify the observation to known poses. As a result, the robot is only able to acquire prior knowledge (templates) or fitness models (statistical classifiers), but unlikely to be able to parse clothes into semantic primitives. Willimon et al. [Willimon et al., 2012] proposed a 3D deformable surface estimation approach using energy minimisation. They defined four types of energy terms: smoothness, correspondence, depth and boundary terms. These terms are the constraints that prevent the garment's surface from deviating from the canonical configuration. As an extension [Willimon et al., 2013b], they removed the SURF feature correspondence term and extended the boundary term by considering the garment's corners. This method is capable of generating a mesh for the deformable surface, but the generated mesh is constrained to the prior knowledge of its canonical template.

2.6.5 Interactive Perception

Interactive perception is of critical importance in visually-guided clothing manipulation. Through interactive perception, the robot is able to avoid getting stuck in an unrecognisable state and perception confidence can be updated. There exists some interactive perception-based work that has successfully solved some clothing manipulation problems [Cusumano-Towner et al., 2011, Doumanoglou et al., 2014b, Li et al., 2015a, Sun et al., 2015, Willimon et al., 2013a, 2011a]. Specifically, Willimon et al. [Willimon et al., 2011b] first proposed to recognise the clothing's category from hanging configurations. In his approach, the hanging garment is interactively observed as it is rotated. In Cusumano et al. [Cusumano-Towner et al., 2011], in order to bring the garment into an unfolded configuration, the hanging garment is slid along the table edges iteratively until the robot can recognize its configuration. Subsequently, in Doumanoglou et al.'s unfolding work [Doumanoglou et al., 2014b], an active forest is employed to rotate the hanging garment to a recognisable field of view. Li et al. [Li et al., 2015a] proposed a more straight-forward unfolding approach based on pose estimation [Li et al., 2014a,b] through interactively moving the grasping point towards the target positions (e.g. elbows). Moreover, interactive perception has been used in heuristic-based generic clothing manipulation. In Willimon et al. [Willimon et al., 2011a] and our previous work [Sun et al., 2013, 2015], a perception-manipulation cycle is adapted to track the state of the garment and heuristic manipulation strategies are used to unfold and flatten



(a) Depth map captured by AUSA Xtion Pro. (b) Depth map captured by stereo head.

Figure 2.15: The comparison between depth data produced by kinect-like camera and stereo head.

the garment on the table.

2.7 Discussion

This section summarises the limitations of the state-of-the-art of clothes perception and manipulation and how to advance it. In Section 2.7.1, the limitations the the state-of-the-art are illustrated form four aspects: limitation in depth sensing, in visually-guided manipulation, in clothes recognition and in interactive perception. Then these limitations are summarised in Section 2.7.2, and Section 2.7.3 presents how this thesis advances these limitations.

2.7.1 The Limitations of the State-of-the-Art Clothes Perception

Limitations in Depth Sensing

From the investigation reported in Section 2.2, Kinect-like cameras are mostly used in clothes perception and manipulation as well as in robotic manipulation. Although Kinect-like cameras have a good trade-off between price, quality and speed, the resolution and quality of depth sensing is still limited for the tasks of clothes manipulation. Take flattening for an example: the small wrinkles on clothes is approximately 0.5 cm or less, whereas the accuracy of Kinect-like cameras is around 0.5 cm. As a consequence, small wrinkles are unlikely to be captured using Kinect-like cameras (shown in Fig. 2.15).

Limitations in Visually-Guided Clothes Manipulation

The above survey shares common ground in terms of the visual perception capabilities of the robot: they are constrained to a specific garment or task at hand. In other word, the state-of-the-art visual perception approaches for clothes manipulation are not generic enough for more than one task. To the best of the author's knowledge, Ramisa et al. [Ramisa et al., 2013] proposed a 3D descriptor that is exploited in clothes grasping, wrinkle detection, and category recognition tasks, which is the only generic approach for multiple clothes perception tasks. Nevertheless, their approach shows two main limitations: Firstly, their proposed approach simply adapts all tasks into a *black-box classification* problem rather than parse the clothes appearances into details. Secondly, validations within real-life environments have not been provided with respect to garment grasping and category sorting.

Limitations in Clothes Recognition

The most commonly reported clothes recognition approaches [Kita et al., 2004, 2009a,b, Li et al., 2014a,b, Willimon et al., 2013a, 2011b] are devised for recognising garments from hanging configurations, since the configuration space is greatly reduced. In these scenarios, a large robot with a large working space is required. Hence, medium sized robots cannot be used to manipulate adult garments. As free-configuration clothing typically presents several occlusions and a much larger configuration space, the performance of existing simulated approaches within these scenarios is limited. Although recognising clothes categories from free-configurations has been explored by [Ramisa et al., 2013, Willimon et al., 2011b], the performance of their reported methods are very limited and no real-robot experiments are provided.

Limitations in Interactive Perception

In interactive perception, the most critical problem is to describe the confidence of the perception as it is treated as the state of perception and also can be used as the halting criteria of interactive-perception cycles. In general, this confidence is given by the prediction or registration error.

Researchers have proposed various feature representations for clothing visual perception problems. However, few inference (classification) methods have been investigated. Most of them use Support Vector Machines (SVM) and Random Forests as the classifier [Doumanoglou et al., 2014a,b, Li et al., 2014a,b, 2015a, Ramisa et al., 2012, Willimon et al., 2013a, **?**], and in some earlier work, K-nearest neighbour (kNN) is used [Willimon et al., 2011b]. Classifiers in the SVM family classifiers do not provide confidence in their predictions but instead provide

a hard decision. Forest-like classifiers [Doumanoglou et al., 2014b, Willimon et al., 2013a] can generate the confidence from voting, but the reliability of such estimates is limited by the number of trees and has no formal probabilistic basis. Besides the classification-based approaches, non-linear registrations are also widely used to match the visual perception with known templates [Cusumano-Towner et al., 2011, Kita et al., 2004, 2009a,b, Li et al., 2015a]. Registration-based methods are capable of matching hanging or sliding-table-edge configurations and the matching errors can be adapted as the measurement of confidence. However, the performance of registration is more sensitive to the complexity of the garment configurations, which means they are unlikely to be able to match the configurations when subject to high occlusion e.g. on-table configurations.

2.7.2 Summary

Overall, from the literature review, the conclusion can be given that the current state-of-the-art cloth manipulation usually focuses on specific tasks rather than the basis of understanding generic cloth configurations. The key limitations of the state-of-the-art clothes visual perception and visually-guided manipulation can be summarized as:

- Existing methods for visual perceptions of clothing are devised for specific tasks where the 3D configuration is not sufficiently understood.
- The depth sensing of robot clothes manipulation is mainly based on Kinect-like cameras, the accuracy of which is not enough for dexterous garment manipulation. The existing garment configuration understanding is not enough for performing dexterous manipulation.
- The majority of the reported visually-guided clothes manipulation solutions are not following perception-manipulation cycles. As a consequence, the robot is unlikely to be able to recover the garment from ill-posed configurations.
- The predominant reported visual interpretation of 3D garment configurations uses either preliminary global representations (e.g. polygons) or pale and black-box local-based representation (e.g. SIFT, BoF). The features are not hand-crafted to describe the attributes which are important to the classification problem. The existing clothes category recognition from free configuration methods are of a very limited performance, which barely have the potential for industrial application.
- In predominant reported clothes recognition pipelines, the classifiers e.g. SVM, RF, kNN are non-probabilistic discriminative classifiers and cannot provide the confidence of prediction.

2.7.3 How to Advance

Corresponding to these limitations, this thesis provides the following solutions to advance the state-of-the-art:

- This thesis proposes a generic visual architecture to provide a bottom-up sufficient parsing and interpretation of 3D clothes configuration for multiple laundering tasks, which is capable of grasping, flattening and recognizing garments.
- In this thesis, high-resolution and high-accuracy stereo head data is used instead of Kinect-like camera data, which provides precise depth sensing for dexterous clothes manipulation. In terms of garment sensing, a precise garment 3D configuration parsing approach is devised that localises and quantifies the landmarks of the garment's shape-topological surface.
- The proposed dual-arm flattening approach employs perception-manipulation cycles, interoperating 3D precise configuration understanding and generic manipulation strategy, which can track task status and recover from ill-posed configurations or incorrect observations and operations.
- Instead of using impoverished representations designed for specific tasks, this thesis proposes a sufficient representation, fusing statical and vocabulary features non-linearly, extracted from generic 3D landmarks and surfaces, to describe the 3D garment configuration.
- Gaussian Process multi-class classification is adapted for predicting the category of unknown clothing. The predictive probabilities generated by GP are adapted into a novel interactive sorting pipeline which lead to substantial improvements on the recognition performance.

Chapter 3

Clothes Manipulation Evaluated in Physical Simulation

This chapter investigates the contribution of visual perception in the robotic manipulation task of flattening wrinkled garments by eliminating visually detected wrinkles. This task requires dexterous manipulation with precise visual perception because appropriate flattening actions are likely to eliminate the wrinkles effectively whilst inappropriate actions are likely to generate more wrinkled configurations.

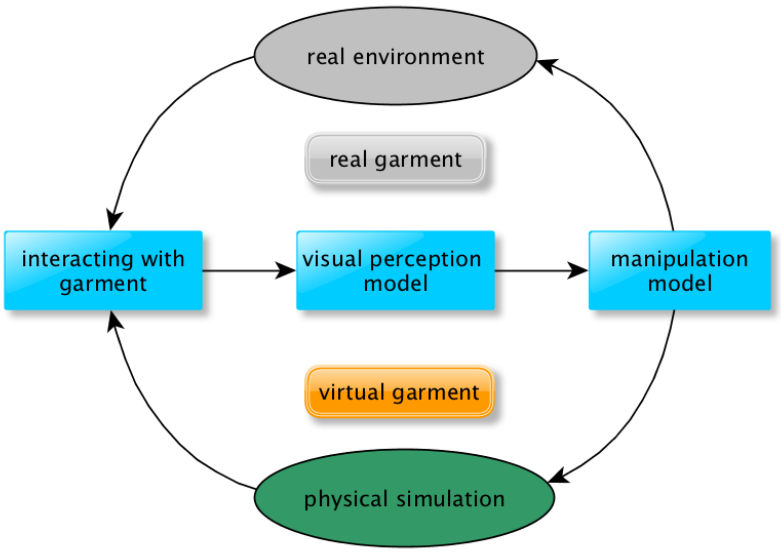


Figure 3.1: The general framework of visually-guided clothes manipulation.

A general framework of visually-guided clothes manipulation is shown in Fig. 3.1. There exist two ways to verify the performance of visually-guided manipulation: one is from simulated experiments and the other is from the statistics of real robot experiments. This chapter evaluates the performance of different visual perceptions in terms of single-arm flattening in

a repeatable and controlled physical clothes simulation.

3.1 Introduction

In this chapter, a hand-eye interactive virtual robot manipulation system that incorporates a clothing simulator is developed to close the garment visual sensing and interaction cycle. This chapter presents the technical details and compares the performance between two visual perception methods in terms of detecting, representing and interpreting wrinkles within clothing surfaces captured with high-resolution depth maps. The first proposed method relies upon a clustering-based method for localising and parametrising wrinkles, while the second method adopts a more advanced geometry-based approach in which shape and topology analysis underpin the identification of cloth configurations (e.g. wrinkles). Having interpreted configurations of clothes by means of these methods, a heuristic-based flattening strategy is then executed to infer the appropriate forces needed to flatten perceived wrinkles. A greedy approach, that attempts to flatten the largest detected wrinkle for each perception-iteration cycle, has been successfully adopted in this work. This work presents the results of the proposed heuristic-based flattening methodology cooperating with clustering-based and geometry-based features, respectively. The experiments indicate that the geometry-based feature has the potential to provide a greater degree of clothing configuration parsing and thus improves flattening performance.

In section 3.2, the motivation and objective are given. In Section 3.3, the virtual cloth manipulation system for evaluating the flattening performance is introduced. Section 3.4 and Section 3.5 present two different feature extraction approaches, respectively. In Section 3.6, the heuristic single-arm flattening strategy is introduced, and experiments are presented in Section 3.7. Finally, this chapter is concluded in Section 3.8.

3.2 Motivation and Objectives

Deformable clothing can be classified according to their elasticity, viscosity, plasticity and resilience. Clothing is therefore the most challenging deformable object for perception and manipulation tasks [Luible, 2008]. These characteristics of clothes make the clothes configurations extremely sensitive to the external forces generated by the manipulator; therefore, interacting with clothing requires dexterous manipulation. In the literature, a range of vision-based approaches are proposed as the solution for each procedure of the autonomous laundering pipeline. However, none of them exactly investigates the contribution of visual perception and evaluates how much the manipulation will be effected by an advanced or a preliminary visual perception.

The required perceptual and hand-eye coordination capabilities come naturally to humans but pose a challenge to current autonomous robotic systems. In order to evaluate clothes manipulation performance, a hand-eye interactive virtual robot manipulation system is developed, which provides a virtual hand-eye coordination environment without including the calibration error and physical interaction faults. The virtual robot manipulation system incorporates a physics based simulator, simulating mass-spring model-based cloth, the frictions between clothes and table, the interactions between the robot and the cloth, and a virtual camera system capturing depth maps of the manipulation platform. Moreover, this system can integrate different visual feature extraction and robot manipulation methods. This chapter integrates different feature extraction approaches and the same heuristic flattening strategy; thus the utilities of different visual perceptions can be directly recognised from the performance of manipulation.

The objective of this chapter is to investigate the contribution of visual perception in terms of parsing the underlying structures of clothing for visually-guided manipulation task (flattening). For this purpose, two visual perception methods for parsing the clothes configurations from depth maps are used: (a) the clustering-based wrinkle analysis, where the local and global discontinuity statistics are calculated (Section 3.4), and (b) the geometry-based wrinkle analysis, employed to extract a topological representation of the cloth (Section 3.5). To demonstrate the performance of these visual perception techniques, the cloth-flattening task, a key step in which is the localisation and quantification of wrinkles, is considered for verifying different visual perceptions. The experiments are conducted within a physical simulator in which the performance of the different perception systems with identical cloth initial configurations are compared. This geometry-based wrinkle analysis methodology has the potential to outperform the more preliminary clustering-based approach in terms of manipulating, and specifically, flattening clothing.

3.3 Virtual Clothes Perception and Manipulation System

In order to explore and validate visually guided clothing manipulation, a virtual hand-eye interactive manipulation system is developed. The physical cloth simulator enables us to apply the required actuation forces accurately while avoiding sources of error such as the robot's joints and noise introduced by sensing systems. In this chapter, physics-based virtual cloth simulation is employed to evaluate the flattening performance of different feature extraction methods. This simulated approach can reproduce the same initial cloth configurations for each experiment, which of course would be difficult to achieve if real cloth were to be used for this purpose. It must be emphasised that the presented feature extraction approaches are

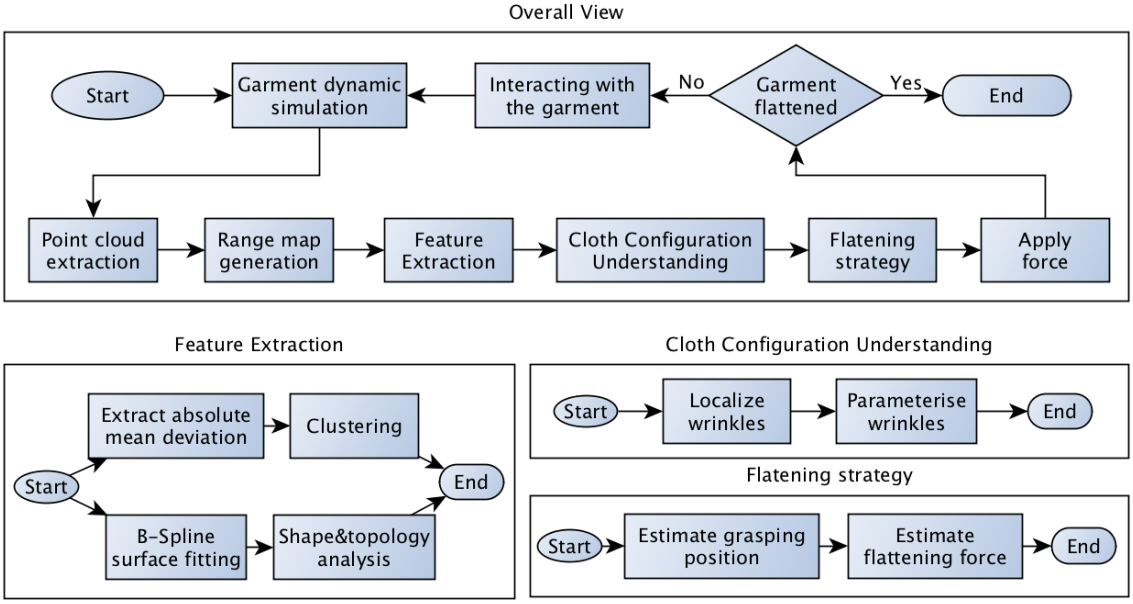
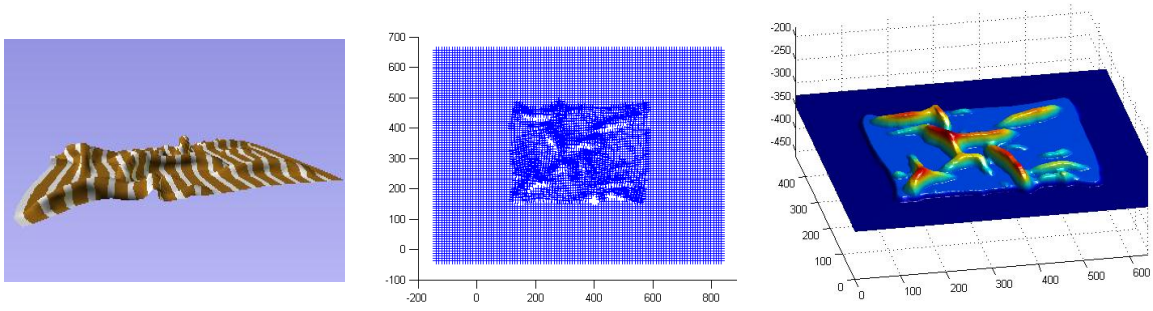


Figure 3.2: Flow chart of the virtual cloth manipulation system. The system begins with initialising the cloth in the physical simulation. When the simulated cloth becomes static, the point cloud can be obtained from cloth particles. Thereafter, a virtual depth camera is used to capture a depth map of the scene. The visual features are then extracted by the means of two proposed methods, and all wrinkles are detected and quantified, thus cloth configuration is understood. The flattening strategy indicating the location of grasping, direction of flattening, and the magnitude of the force is inferred from the parsed configuration. Before applying a flattening force, the status of the cloth is checked to see whether its 'flatness' meets the halting criteria. If yes, the process is terminated; otherwise flattening is acted on the garment and the dynamic interaction between garment and external environment is simulated.

not only able to work in virtual simulation but also in real robot practice using real-world RGB-D data (such as Kinect-like camera data or stereo data). Examples of feature extraction based on real-world Kinect data and the demonstration of autonomous single-arm flattening are presented in Fig. 4.11.

The schema of the virtual cloth manipulation system is illustrated in Fig. 3.2. This virtual system follows the perception-manipulation cycle: the processing cycle starts by either initialising cloth with specific wrinkled configurations (presented in Section 3.7.1) or applying a computed force to deform the cloth. Once the cloth has attained static equilibrium, a point cloud is generated from all the particles that compose the cloth (perception). A range map is then generated from the point cloud (an example is shown in Section 3.3), which is passed to the feature extractors as described in Sections 3.4 and 3.5. Following analysis of the cloth scene, an action (force) is inferred by the flattening strategy (presented in Section 3.6) and applied to flattening the largest wrinkle. This processing cycle is iteratively applied to generate an animation depicting the cloth surface during the flattening process.



(a) Simulated cloth (b) Point cloud (c) Range map

Figure 3.3: The demonstration of capturing a depth image in the proposed virtual clothes manipulation system. (a) Cloth rendered in virtual simulation. (b) Generated point cloud. This point cloud is composed of 2475 particles (a square cloth of 54 $\times$ 45 particles). (c) Computed range map from the point cloud.

Cloth Simulation

The simulated virtual cloth is composed of particles that are governed by structural, shear and bending constraints [Lander, 1999]. The motion of these particles is modelled in terms of Newton's laws of motion and a mass-spring model, as described in [Lander, 1999, Provot, 1995], operating under gravity. For each of the above constraints, the interaction of each particle is further restricted by an offset that limits the range of distances between all connected particles (i.e. to prevent particles from getting too close or far away from each other). The system has also incorporated the frictional forces exerted by a virtual table acting on the cloth. For this purpose, Ridson's friction model is adapted (as detailed in [Bridson et al., 2002]), which includes both static and sliding frictional forces.

The reported simulator is implemented in Visual Studio 2010 (C++) running in Windows 7, 64-bits. OpenGL [Shreiner et al., 2009] is used to render the 3D surface of the cloth. This simulation can be initialised by loading 3D cloth points and their spring constraints. In this case, the cloth initial configurations can be reproduced, which is required in comparison experiments. The processing time consumed during each cloth flattening simulation iteration is approximately 10 seconds, depending on the magnitude of the forces acting on the cloth.

From Point Cloud to Range Map

A range map comprises a matrix of range values, in which each pixel position stores the distance from the perspective centre of a camera to a location on the surface of the observed scene (range values exhibit the central projection geometry of standard RGB images). Accordingly, a range of standard 2D image processing algorithms can be directly applied to the 2D matrix structure of the range map [Bowyer et al., 2006]. Moreover, a range map can save

$2/3$ of storage space as compared to a point cloud. Therefore, in this research, range map representation is adapted over the unstructured point cloud.

Fig. 3.3 illustrates the overall pipeline developed to capture a depth image from the simulated virtual cloth. In order to generate a surface from the point cloud, a cubic convolution interpolation algorithm [Keys, 1981] is employed to generate a 2.5D range map (Fig.3.3-c). As the depth sensing of the proposed virtual manipulation system is targeted to be as realistic as acquiring range maps from a stereo vision system [Cockshott et al., 2012b], or more general RGB-D sensors, surface points self-occluded from the camera's point-of-view on the simulated world are deleted by means of a hidden point removal algorithm [Katz et al., 2007]. This step removes points that can potentially affect the surface construction and interpolation process. In this work, the camera's point of view is selected to be perpendicular to the table plane; this approximates the perspective from which a robot might observe a garment lying on a table.

3.4 Baseline Perception – Clustering-Based Wrinkle Analysis

Absolute Mean Deviation Feature

A wrinkle is a basic folding configuration exhibited by cloth that is defined as a statistical discontinuity in the range values representing the cloth surface. This approach also defines wrinkles to be approximately linear structures with definite start and end points. In order to detect wrinkles, a 'wrinkledness' or 'wrinkle strength' score is estimated by computing the local average absolute deviation of range values in square patches on the range map (three different square sizes are adapted; more details are shown in Fig. 3.4). More specifically, for each patch p_i, the absolute deviation between the patch range and cloth range is calculated by:

$$amd_{p_i} = \frac{\sum_{N_j} \mid d_j - d_c \mid}{N_j}, \tag{3.1}$$

where N_j is the number of pixels in this patch, d_j is the depth value of the jth pixel, and d_c is the mean depth value of the cloth. In the implementation of this thesis, absolute deviation features computed with different patch sizes are pixel-wisely merged together into a single feature strength map. A Gaussian smoothing process is then applied to obtain the final feature map that preserves continuity across pixels.

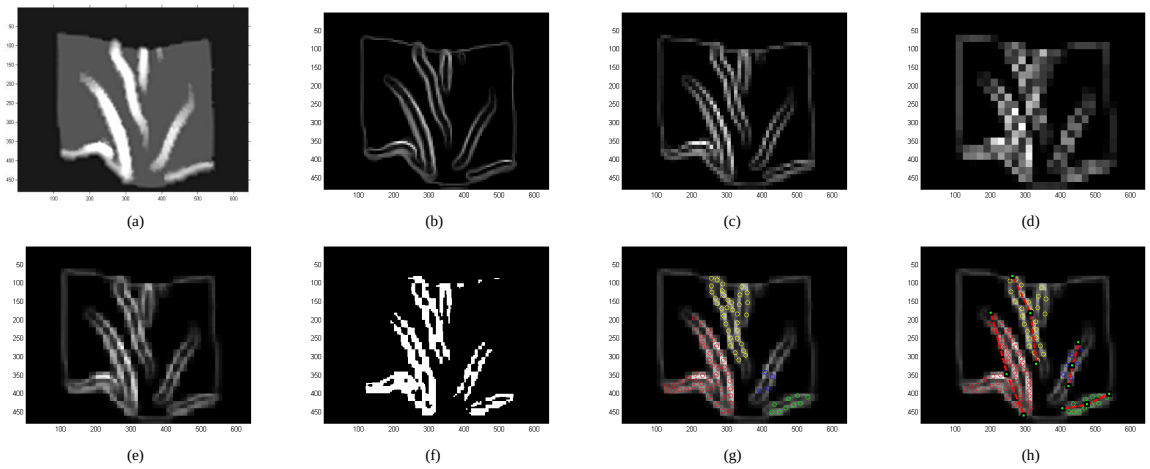


Figure 3.4: An example of clustering-based wrinkle detection and parametrization. In (a), the range map of wrinkled cloth is shown. In the implementation, mean absolute deviation features are computed with patch sizes of 5, 10, 20 as shown in (b), (c) and (d), respectively. (e) shows the final merged feature map with Gaussian smoothing. The previous three deviation maps are merged by simply averaging. In (f), the pixel-level segmentation using threshold σ_1 is shown. In (g), the round dots represent the cluster centres of K-means. Then, the clusters are grouped to different wrinkles through hierarchical clustering as shown in different colors. In (h), the detected wrinkles are shown, in which the red one is the largest one. In the implementation of this work, σ_1 is set at 0.5, σ_2 is set at 35, and N_{kmeans} is set at 100. From empirical investigation, these thresholds work well in practice.

Clustering-Based Wrinkle Description

Having computed the absolute mean deviation feature in the previous section, a K-means clustering algorithm is applied to the x-y coordinates of those pixels labelled as highly wrinkled. Thereafter, these clusters are jointed using a bottom-up hierarchical clustering algorithm [Johnson, 1967] in order to group them into salient wrinkles. The two end points of a wrinkle, and also the wrinkle centre, can be computed from the final clustering (as shown in Algorithm 1). A whole process of feature extraction is demonstrated in Fig. 3.4.

3.5 Advanced Perception – Geometry-Based Wrinkle Analysis

Section 3.4 presents a clustering-based wrinkle detection approach. That approach has the following immediate limitation: The clustering-based detection method lacks an effective wrinkle separation mechanism. When the cloth is highly wrinkled, the patch used to measure local range variance will overlap adjacent wrinkles and therefore tend to merge these together. As a consequence, the clustering-based detection and representation is unable to

Algorithm 1 Clustering-Based Wrinkle Construction Algorithm.

1: **In:** the average absolute deviation map D, a threshold to distinguish wrinkle pixels σ_1, and a distance threshold for the hierarchical clustering algorithm, σ_2. The number of k-means clustering centres N_c.
2: **Out:** the largest wrinkle $w_{largest}$ (its length l and centre c, endpoints p_{start}, p_{end}).
3: k-means clustering algorithm is applied on the x-y coordinates of pixels that satisfy $D > \sigma_1$, in order to obtain N_c clusters $\{c_1, \cdots, c_{100}\}$.
4: Record the number of pixels in each cluster $\{n_1, \cdots, n_{100}\}$.
5: Cluster the k clusters $\{c_1, \cdots, c_{100}\}$ in the last step using a Hierarchical Clustering algorithm (from bottom to top), and the clustering will terminate if the distance is larger than a threshold σ_2, thereafter, get new clusters(wrinkles) $\{w_1, \cdots, w_n\}$ and the corresponding number of pixels $\{N_1, \cdots, N_n\}$
6: Get the cluster $w_{largest}$ with the largest number of pixels in $\{N_1, \cdots, N_n\}$.
7: The length $l = max(|(x_{max}, y) - (x_{min}, y)|, |(x, y_{max}) - (x, y_{min})|)$ and the endpoints are those which have the lager distance, x_{max}, x_{min} are $x - y$ coordinates in w_i that has the maximum and minimum coordinate in the x axis, while y_{max}, y_{min} are the maximum and minimum for the y axis. $c = mean_{p \in w_{largest}}([x_p, y_p])$.

localise wrinkles spatially and quantify their magnitude with sufficient accuracy to support reliable dexterous cloth manipulation.

In order to advance the state-of-the-art in the analysis of deformable objects (i.e clothing), a more advanced geometry-based feature extraction approach is needed, which parses the cloth surface hierarchically from low-level curvature features, to middle-level topology features, and finally to high-level wrinkle descriptors. This process comprises four steps: *B-spline surface fitting*, *cloth shape and topology analysis*, *wrinkle construction*and *wrinkle measurement*.

As geometry-based features such as curvature and shape index [Koenderink and van Doorn, 1992] are susceptible to high frequency noise, a B-Spline surface fitting approach is employed (more details are shown in Appendix A). Then the surface topographic features (including ridges, wrinkle contours, and surface shape types) are detected by computing the shape index of the cloth surface. Thereafter, wrinkle structures are constructed and represented by fitting polynomial curve models. The wrinkles are quantified through a triplet of values (detailed in the next section) that allow the location, length, height, width and energy. Finally, based on this quantification, a flattening heuristic is applied to reduce the magnitude of the largest wrinkle detected.

Clothes Topology and Shape Analysis

In order to detect wrinkles on the clothes surface, this method first provides a geometric definition of a wrinkle (Definition 1 below). While the predominant surface topology information is encapsulated in the ridges, the wrinkle's contour defined by the boundary of concave

and convex surfaces (Definition 2 below) is also important especially when parametrising wrinkles. Since the convex ridge defining the wrinkle is surrounded by a concave valley, this convex/concave boundary is used to segment the wrinkle. In this chapter, the definition of a ridge is the same as that given by [Belyaev and Anoshkina, 2005]. The surface shape categories are classified into 9 types of surfaces using shape index [Koenderink and van Doorn, 1992].

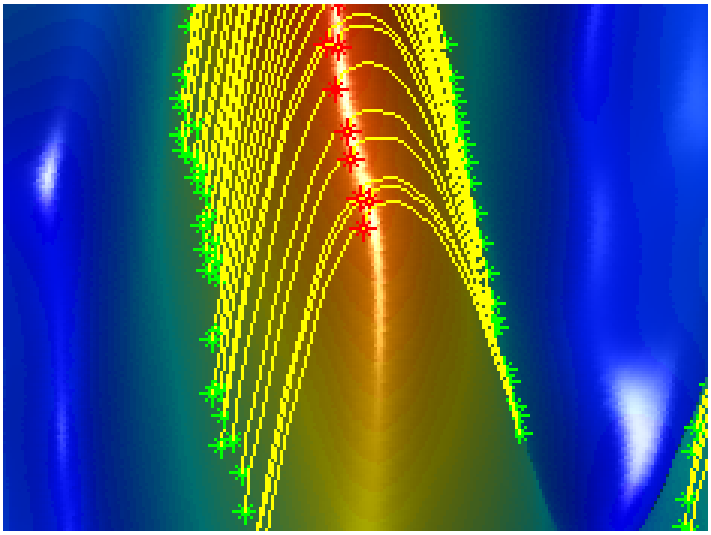


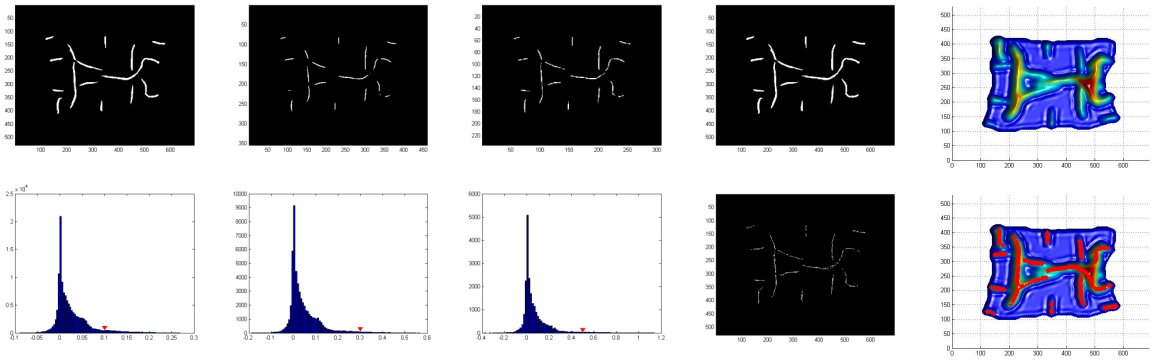
Figure 3.5: Representation of a wrinkle using triplets. A close-up example of triplets is shown, in which each ridge point (red) is matched with its two corresponding contour points (green).

- **Definition 1.** A wrinkle comprises a continuous ridge line contained within in a region where the surface shape type is 'ridge'. The wrinkle (ridge line) is delimited (bounded) by two contour lines, each located on either side of the maximal curvature direction.

- **Definition 2.** Ridge points are defined at the positive extrema of maximal curvature in a range map. The wrinkle's contour is defined by the boundary of the concave surface surrounding the convex ridge on which each ridge point is located.

- **Definition 3.** A wrinkle can be quantified by means of triplets comprising a ridge point and the two wrinkle contour points located on either side of the ridge, along the maximal curvature direction (as shown in Fig. 3.5).

The positive extrema of maximal curvature (ridges) are detected by thresholding, and in order to detect wrinkles of different magnitudes, the ridges are detected at different scales. More specifically, given a range map I, a Gaussian Pyramid [Cyganek and Siebert, 2011] of three layers $\{\varphi_{L1}, \varphi_{L2}, \varphi_{L3}\}$ is constructed. The pyramid is computed iteratively by applying a



(a) First Layer (b) Second Layer (c) Third Layer (d) Non-maximum suppression (e) Cloth surface and ridges

Figure 3.6: The process of multi-scale ridge detection with non-maximum suppression. Top images (a), (b) and (c) illustrate ridge line detections over different scales. In the implementation of this work, the thresholds of k_{max} in (a), (b) and (c) are 0.1, 0.3 and 0.5, respectively. The histograms of k_{max} curvature are shown in the bottom images of (a), (b) and (c), in which their thresholds are marked as red triangles. In (d), the top figure shows the raw ridges merged over three scales, while the bottom image presents the final ridges after non-maximum suppression. The final result of ridge line detections is shown in the bottom image of (e).

Gaussian low-pass filter ($\sigma = \sqrt{2}$) to the current pyramid level and then reducing the current resolution by a sub-division factor of $\sqrt{2}$. For each point p in this pyramid $\{\varphi_{L1}, \varphi_{L2}, \varphi_{L3}\}$, the mean curvature C_m^p and Gaussian curvature C_g^p are first calculated by Eq. 3.2 and Eq. 3.3, respectively:

$$C_m^p = \frac{(1+(f_y^p)^2)f_{xx}^p + (1+(f_x^p)^2)f_{yy}^p - 2f_x^p f_y^p f_{xy}^p}{2(\sqrt{1+(f_x^p)^2+(f_y^p)^2})^3} \tag{3.2}$$

$$C_g^p = \frac{f_{xx}^p f_{yy}^p - (f_{xy}^p)^2}{(1+(f_x^p)^2+(f_y^p)^2)^2} \tag{3.3}$$

where the first derivatives (f_x^p and f_y^p), and second derivatives (f_{xx}^p, f_{yy}^p and f_{xy}^p) are computed by means of a Gaussian convolution[1] rather than by computing the central differences. An alternative way is to compute gradients from B-Spline parameters, which is introduced in Appendix A. The maximal and minimal curvature at point p can then be calculated using C_m^p and C_g^p as follows:

$$k_{max}^p, k_{min}^p = C_m^p \pm \sqrt{(C_m^p)^2 - C_g^p}. \tag{3.4}$$

The positive extrema of maximal curvature is detected in each scale $\{\varphi_{L1}, \varphi_{L2}, \varphi_{L3}\}$ by setting different thresholds for k_{max}. These thresholds are selected by analysing the histograms of k_{max} (shown in Fig. 3.6). The final raw ridge map R_{raw} is obtained by merging the ridges detected in $\{\varphi_{L1}, \varphi_{L2}, \varphi_{L3}\}$ by pixel-wise addition of these maps. As it can be observed in

[1] In the implementation of this proposed work, σ is 0.8 and the template size is 7×7.

the upper images in Fig. 3.6(d), the ridges in R_{raw} are wide and coarse. In order to obtain a finer ridge map R in which the width of the ridge lines is only one pixel, a Canny-like non-maximal suppression algorithm is applied. Since R is a binary map, and therefore does not contain magnitude values, the range values are used to quantify the edge magnitude values. Moreover, as the purpose is to retain the maximal value along the maximal curvature direction, the maximal curvature direction θ (Eq.3.5) is adapted instead of a gradient direction as in the Canny edge detector.

$$\theta = \tan^{-1} \frac{\nabla_y k_{max}}{\nabla_x k_{max}}. \tag{3.5}$$

In Equation 3.5, $\nabla_y k_{max}$ and $\nabla_x k_{max}$ are the gradients of k_{max} on y and x directions, computed by Gaussian convolution (the same method used in Eq. 3.3). A more detailed description of the non-maximum suppression applied is given in Algorithm 2.

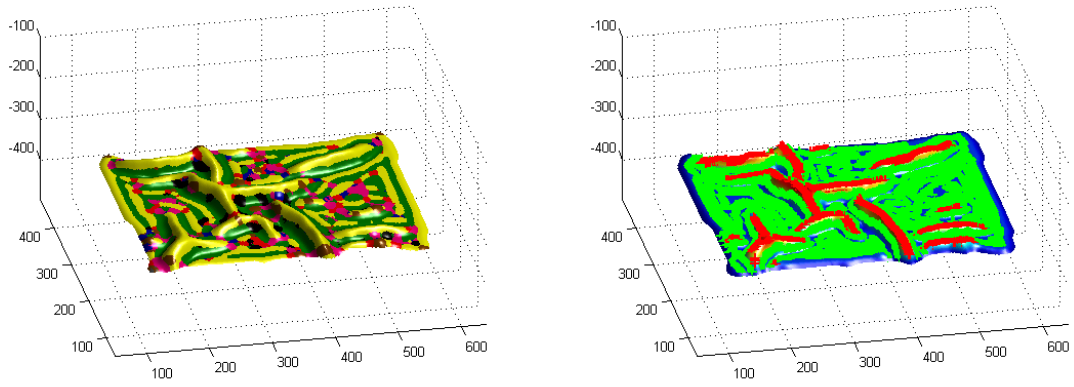
Algorithm 2 Ridge non-maximum suppression.

1: **In:** The range map I and all ridge points in the raw ridge map $R_{raw} = \{p_r^1, \cdots, p_r^N\}$ detected from I.
2: **Out:** The refined ridge map $R = \{p_r^1, \cdots, p_r^n\}$.
3: Construction Gaussian pyramid $\{\varphi_{L1}, \varphi_{L2}, \varphi_{L3}\}$
4: Initialise R is empty.
5: **for** each ridge point p_r^i in the raw ridge map R_{raw} **do**
6: Compute the direction of maximal curvature θ using Eq. 3.5 in scale φ_{L3}.
7: Discrete θ to 8 uniform directions (north, south, west, east, northwest, northeast, southwest, southeast).
8: Check the two neighbours along θ direction whether p_r^i has the largest depth value.
9: **if** (**then**p_r^i has the largest depth value)
10: Add p_r^i into R.
11: **end if**
12: **end for**

Shape index [Koenderink and van Doorn, 1992], produces a continuous shape types map S classifying local shape of surface regions into a real-valued index value in the range [-1, 1]. The shape index value is quantised into 9 intervals corresponding to 9 surface types – cup, trough, rut, saddle rut, saddle, saddle ridge, ridge, dome and cap. Among all the shape types, 'ridge' is critical in the analysis and description of wrinkles. The shape index value S^p of point p can be calculated as follows [Koenderink and van Doorn, 1992]:

$$S^p = \frac{2}{\pi} \tan^{-1} \left[\frac{k_{min}^p + k_{max}^p}{k_{min}^p - k_{max}^p} \right], \tag{3.6}$$

where k_{min}^p and k_{max}^p are the minimal and maximal curvatures at point p computed using Eq. 3.4. In order to parse the shape information exhibited by the visible cloth surface, the shape index is calculated from the range map and majority rank filtering is applied (an example is



(a) In this figure, the 'ridge' region is plotted as yel- low color, which is highly instrumental for the wrinkle grouping and triplet matching. In the implementation of this work, the radius of the majority ranking filter is set to 5 pixels.

(b) The topology of a clothing surface, where red points refer to ridges and green refer to wrinkle's contour.

Figure 3.7: The shape index and topologies map on B-Spline fitted surface.

shown in Fig. 3.7-b). This non-linear filtering removes outlier surface classifications and can be tuned to produce a relatively clean classification of topology types over the cloth surface.

Nine types of surface shape are labelled, four of which are convex (saddle ridge, ridge, dome, cap) and the others are concave (cup, trough, rut, saddle rut, saddle). Having estimated the shape index of the cloth's surface, the boundary of convex and concave surfaces can then be obtained. Following detection of the wrinkle's contour, a full description of the cloth topology can be obtained. An example can be seen in Fig. 3.7(a).

Wrinkle Construction

The wrinkle construction process consists of two steps: connecting ridge points into contiguous segments and grouping segments in order to represent wrinkles. In the first step, neighbouring pixels are iteratively connected in the ridge map until contiguous segments are obtained. Specifically, each ridge point is first labelled as an active point, then its eight nearest neighbours are checked to determine whether they are also ridge points. If the latter holds true, they are jointed together, label the new end points to active points and remove the old points. This process will be executed iteratively until the segments no longer grow. After this process, the end points of segments are labelled as 'active' points. This first step is further described in Algorithm 3. The minimal distance l_2 between every two active ridge points constrained in the same region is used as the distance measurement between two segments:

$$dist(s_i, s_j) = \begin{cases} \min \parallel \forall r_m^a \in s_i, \forall r_n^a \in s_j \parallel_2, \ if \ r_m, r_n \in \gamma \\ \infty, \quad otherwise \end{cases}, \tag{3.7}$$

where s_i and sj are two ridge line segments consisting of a range of active ridge points $\{r_1^a, ...r_M^a\} \in R$ and $\{r_1^a, ...r_N^a\} \in R$, and γ is a 'ridge' region in the shape index map.

Algorithm 3 Tracking contiguous ridge points.

In: All ridge points in the ridge map $R = \{p_r^1, \cdots, p_r^N\}$.
Out: Wrinkle segments $S\{s_1, \cdots, s_n\}$.
for Each ridge point p_r^i in R **do**
    Set segment $s_i = \{p_r^i\}$, add p_r^i into s_i's active points Ap_{s_i}
    while s_i does not change **do**
        for each ridge point p_r^j in Ap_{s_i} **do**
            if at least one of p_r^j's eight neighbours $\{\cdots, p^{Nj}, \cdots\}$ are ridge points **then**
                Remove $\{\cdots, p^{Nj}, \cdots\}$ from ridge points $\{p_r^1, \cdots, p_r^N\}$, and add $\{\cdots, p^{Nj}, \cdots\}$ to s_i.
                Remove p_r^i from Ap_{s_i} and add $\{\cdots, p^{Nj}, \cdots\}$ to Ap_{s_i}.
            end if
        end for
    end while
end for

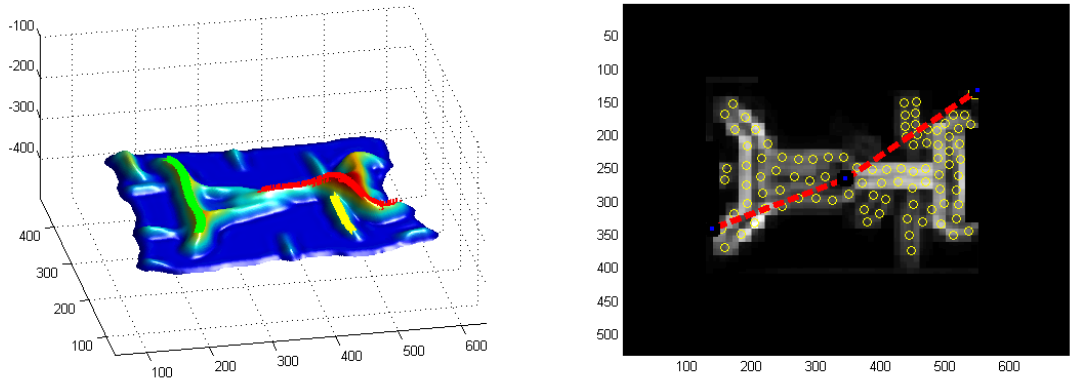
The second step is to group the short segments obtained in the previous step into long wrinkles. This process is achieved by connecting the two closest segments iteratively, in which the distances between every two segments (wrinkles) are measured by Eq. 3.7. If two closest segments are in the same 'ridge' region, then they are grouped into a larger segment. This process is executed iteratively until wrinkles are constructed from ridge segments. More details are shown in Algorithm 4.

Algorithm 4 Constructing contiguous wrinkles from segments.

In: Wrinkle segments $S\{s_1, \cdots, s_n\}$.
Out: Fitted wrinkles $W_{fitted}\{w_f^1, \cdots, w_f^n\}$.
while S does not change **do**
    for every two different segments s_i and s_j in S **do**
        Find the nearest s_i and s_j through Eq.3.7.
        Merge s_i and s_j, let $s_i = \{s_i, s_j\}$, and delete s_j from S.
    end for
end while
for each segment s_i in S **do**
    Principal Component Analysis on s_i.
    Rotate s_i that align its two principal orientations be collinear with the x,y axis.
    Fitted a five order polynomial curve w_i to s_i.
    Rotate w_i back to the original position.
end for

After grouping the segments iteratively, several large wrinkles can be represented as a set of ridge points. In order to filter ridge position estimates and interpolate missing ridge values, for each wrinkle, a least-square-error polynomial curve is fitted along its ridge points, and it

is the coefficients of this polynomial which comprise the final representation of the wrinkle. An example of fitted wrinkles are shown in Fig. 3.8(a).



(a) In the implementation of this research, 5th order polynomial curve fitting is employed to approximate wrinkles. In this example, the top 3 largest wrinkles are marked as red, green and yellow, respectively.

(b) The wrinkle detected by clustering-based approach in the same range map is also presented. The clustering based approach is ineffective when the wrinkles connect together since surface shape is not analysed in this approach.

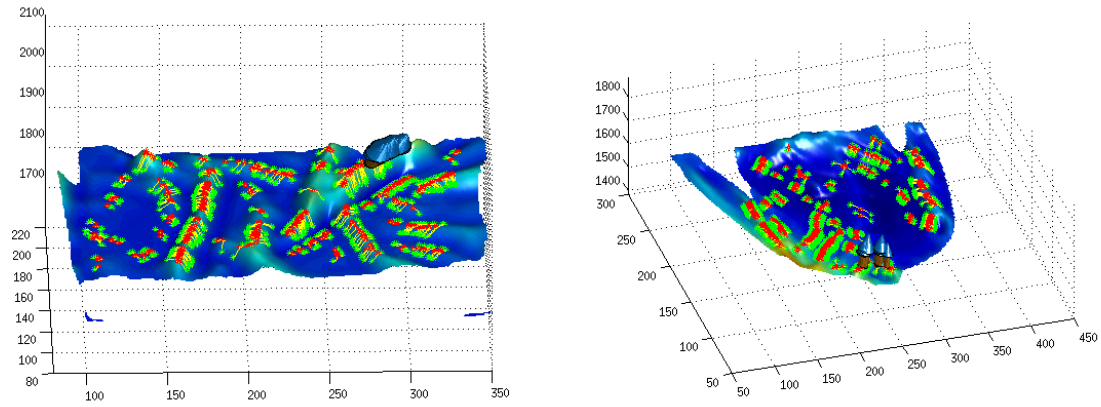
Figure 3.8: Comparison between different wrinkle analysis approaches.

Quantification of Wrinkles

Even though wrinkles can be detected by the proposed cloth shape and topology analysis, and approximated by polynomial curves (as demonstrated in the previous sections), it is not possible to quantify wrinkles since the surface topology classification afforded by the shape index is invariant to the magnitude of the surface variation (e.g. curvature) by definition. In other words, salient and non-salient shapes indicate the same values in shape index. In order to quantify the detected wrinkles, the height and width of a wrinkle are measured in terms of triplets, as defined in *Definition 3*. Accordingly, triplets can also be used as the atomic structure for finding and selecting grasping points (shown in Fig. 3.9). This section describes the construction of triplets by matching ridge points with wrinkle contour points.

As explained in *Definition 1*, the maximal curvature direction θ can be calculated by Eq. 3.5. Given a ridge point p_r in a range map I with scale φ_{L1}, the two corresponding contour points (p_c^l and p_c^r) are searched over the two directions defined by θ and its inverse direction using a depth-based gradient-descent strategy in order to define triplets. During the search process, if the search path traversed is in the same 'ridge' region as p_r (shown as yellow in Fig. 3.7-b), the process will continue. A more detailed description of triplet search is shown in Algorithm 5.

Under ideal conditions, every wrinkle (ridge) point should be matched with its two corresponding wrinkle contour points. Due to occlusions and noise, some wrinkle points (par-



(a) (b)

Figure 3.9: The top-ranking grasping positions detected on real garment data. In these figures, the red and green points refer to ridge points and wrinkles' contour points respectively, and the yellow lines are matched triplets.

ticularly those which do not have two non-empty wrinkle contour points) fail to find their associated contour points and therefore do not generate a triplet. For the robot cloth flattening application, in order to determine the optimal flattening force, the height and width of a target wrinkle must be measured. In this approach, the three points comprising a triplet are used to measure the height and width; hence, only triplets that contain one ridge point and two non-empty wrinkle contour points are regarded as valid primitives for wrinkle quantification. An example of matched triplets is shown in Fig. 3.5. Now, given a triplet t_p containing one ridge point p_r and two wrinkle contour points p_c^1 and p_c^2, the height h_t and width w_t can be calculated as follows:

$$\begin{aligned} h_t &= 2\,\frac{d\,(d-a)\,(d-b)\,(d-c)}{c} \\ w_t &= c, \end{aligned} \tag{3.8}$$

where $a = \parallel p_r, p_c^1 \parallel_2$, $b = \parallel p_r, p_c^2 \parallel_2$, $c = \parallel p_c^1, p_c^2 \parallel_2$, and $d = (a+b+c)/2$. The numerator of the right hand side of the equation is the area of a triangle embedded in a 3D space.

After measuring the height of the wrinkle, the length of a wrinkle can be measured by counting the number of its ridge points, and a wrinkle score ω can then be calculated as follows:

$$score_w = \log(N_r) + \log(\sum_{t_i \in w}^{N_r} h_t/N_r) = \log(\sum_{t_i \in w}^{N_r} h_t), \tag{3.9}$$

where N_r is the number of fitted ridge points in ω, t_i is the ith triplet of ω, and h_t refers to the height of the triplet t_i. This score is a monotonicaly increasing function of $\sum_{t_i \in w}^{N_r} h_t$. Since the force while flattening is positively related to the $score$ of wrinkle (as described in the following section), the $\log$ function is adapted to prevent the score (force) from increasing

Algorithm 5 Triplet Searching Algorithm (left searching).

In: The range maps I in different scales φ_1, $\varphi_{\frac{1}{2}}$, a ridge point p_r.
Out: Triplet structure t_p,
The current point p_l of left searching is set as p_r. $t_p.center = p_r$.
Calculate shape index map S of I in scale φ_1.
Calculate the maximal curvature direction at p θ_p using Eq.3.5.
for $iter = 1; iter \leqslant Max; iter++$ **do**
    while The left searching is unfinished **do**
        if $iter$ == 1 **then**
            The left searching direction $\overrightarrow{d_l}$ is set to θ direction.
        else
            Compute the gradient of p_l's 8 directions in scale φ_1 through calculating the difference in $\varphi_{\frac{1}{2}}$, then $\overrightarrow{d_l}$ is set as the maximal gradient direction.
        end if
        Move p_l one step in φ_1 along $\overrightarrow{d_l}$, then p_l = Move($p_l, \overrightarrow{d_l}$).
        if The path between p_r and p_l is not in the same 'ridge' region in shape index map s **then**
            Set $t_p.left = empty$;
            The left searching is finished;
        else if p_l is wrinkle's contour point **then**
            Set $t_p.left = p_l$;
            The left searching is finished;
        else
            Keeping searching;
        end if
    end while
end for

dynamically (linearly) when the wrinkle's length and height increase.

3.6 Single-Arm Flattening in Simulation

This flattening approach follows a single-arm flattening strategy. The goal is to eliminate the largest wrinkle by exerting a force on the border of the cloth. More specifically, in each flattening cycle, all wrinkles are detected and ranked by Eq. 4.6. The top-ranked wrinkle is then selected and its centre and endpoints are estimated along its principle direction via Principal Component Analysis. The perpendicular bisector of the wrinkle is then calculated from its centre and two endpoints. A force is applied to the intersection point of perpendicular bisector and the cloth border (the closest border with respect to the wrinkle's centre) along the direction of the perpendicular bisector in order to flatten the cloth. It should be noted that, instead of applying a force of fixed magnitude as in [Sun et al., 2013], the magnitude of

the force is now linearly related to the wrinkle's score in this implementation[2]. A simulated flattening example is shown in Fig. 3.13 and the video demo of flattening is available at https://www.youtube.com/watch?v=Rd7hCUZaTx0.

3.7 Experiments

In order to evaluate the effectiveness of the proposed features for parsing the configuration of deformable cloth surfaces, they are embedded into the virtual cloth manipulation system (introduced in Section 3.3). The experimental validation consists of performing flattening experiments with 8 random cloth configurations (shown in Fig. 3.10).

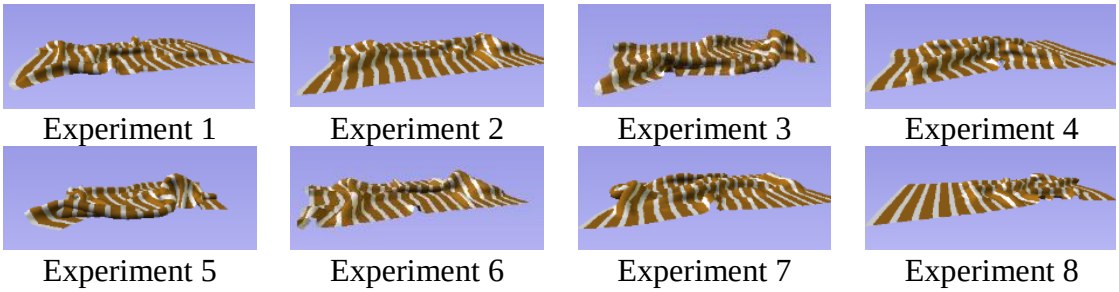


Figure 3.10: Experimental validation of single-arm flattening in simulation. The 8 flattening experiments, which first appear in [Sun et al., 2013], are generated by randomly grasping and dropping the virtual cloth onto a virtual table between 1 and 5 times.

3.7.1 A Benchmark Cloth Flattening Challenge

This work proposes the design of a cloth flattening benchmark challenge based on 8 crumpled cloth states. Any method using the data must conform to the following two constraints:

Constraint 1: During the flattening process, only one force can be applied, which must act along the table ($x - y$) plane. Any flattening using gravity is not permitted.

Constraint 2: The flattening strategy should be generic, without special treatment of cloth corners.

As a result of the above two constraints, manipulation strategies are inferred depending on the perceived state of the cloth and explore the relationship between perception and manipulation. It is also possible to develop generic perception-action loops capable of tackling more complicated manipulation problems based on these constraints. However, the above constraints ensure that the proposed benchmark challenge focuses on the visual parsing and understanding of the cloth's geometrical shape and topology.

[2]Here, a scale factor is used to rescale the wrinkle's score to the magnitude of the flattening force. This scale factor is 20 in this implementation, which is obtained by practical experience.

3.7.2 Global and Local 'Flatness' Indexes

In order to evaluate the performance of the proposed cloth flattening approach, two 'wrinkledness' (or 'flatness') scores are adapted in order to measure how wrinkled the cloth is from 'local' and 'global' perspectives. These measurements are defined as follows.

'Global Flatness': The global flatness measurement provides an estimation of the flatness of the cloth's current state. This is computed in terms of a pixel-level measurement of the stability of the cloth surface as defined in [Ramisa et al., 2012]. Specifically, estimated surface normals are converted to spherical inclination and azimuth angles. The entropy of the bi-dimensional histogram of these angles gives the surface instability quantification. For the implementation of this work, the cloth 'global flatness' is calculated by the average instability value over cloth regions, which is then re-scaled to [100%-0%].

'Local Flatness': As each wrinkle takes up a small proportion of the total cloth area, a high global flatness does not necessarily indicate flatness over all local regions, since the pixel area of a cloth is greater in size than the pixel area of any wrinkle detected on the cloth. This work therefore employs the score of the current largest wrinkle (calculated by Eq. 4.6) in order to measure 'local flatness'. Here 'local flatness' is also re-scaled to [100%-0%].

In experiments of this work, the minimal observed value of 'global flatness' is 0.373. Technically, all surface normals should be [0, 0, 1] in a completely flat case, hence the surface 'instability' measurement (entropy) should also be zero. However, as the cloth has non-zero thickness, the 'instability' measurement along the cloth edges is larger than zero. The value of 0.373 was then obtained by computing the 'global flatness' of a totally flat cloth in simulation. In order to find the maximal value of 'global flatness', the grasp-and-drop experiment is adapted, in which the simulated robot repeatedly grasps and drops the flattest region. The maximal 'global flatness' value obtained after 20 free drops is 3.0127.

The minimum length of ridges for wrinkle fitting is set to 10 units, and the minimum wrinkle height that can be detected is 15 units. In this case, the minimal value of 'local flatness' is $\log(10 \times 10)$ (approximately 4.6, computed using Eq. 4.6). The maximum wrinkle length in simulation is 600 units (the length of the cloth's diagonal) and the maximal wrinkle height is 50 units (also obtained from the grasp-and-drop experiments described above). Therefore the maximal value of 'local flatness' is $\log(600 \times 50)$ (approximately 10.3, computed by Eq. 4.6). It has to be emphasized that, in a real world scenario, physical measurements (e.g. millimetres) are available; however, in simulation, the measurement is a general unit in which the unit is the distance in 3D space of the simulated environment.

In order to track the state of the cloth 'flatness' during the flattening task, both the global and local 'flatness' are recorded at each iteration till no wrinkle can be detected by the proposed feature extraction approach, shown in Fig. 3.11. In Fig. 3.11, the values of the two 'flatness'

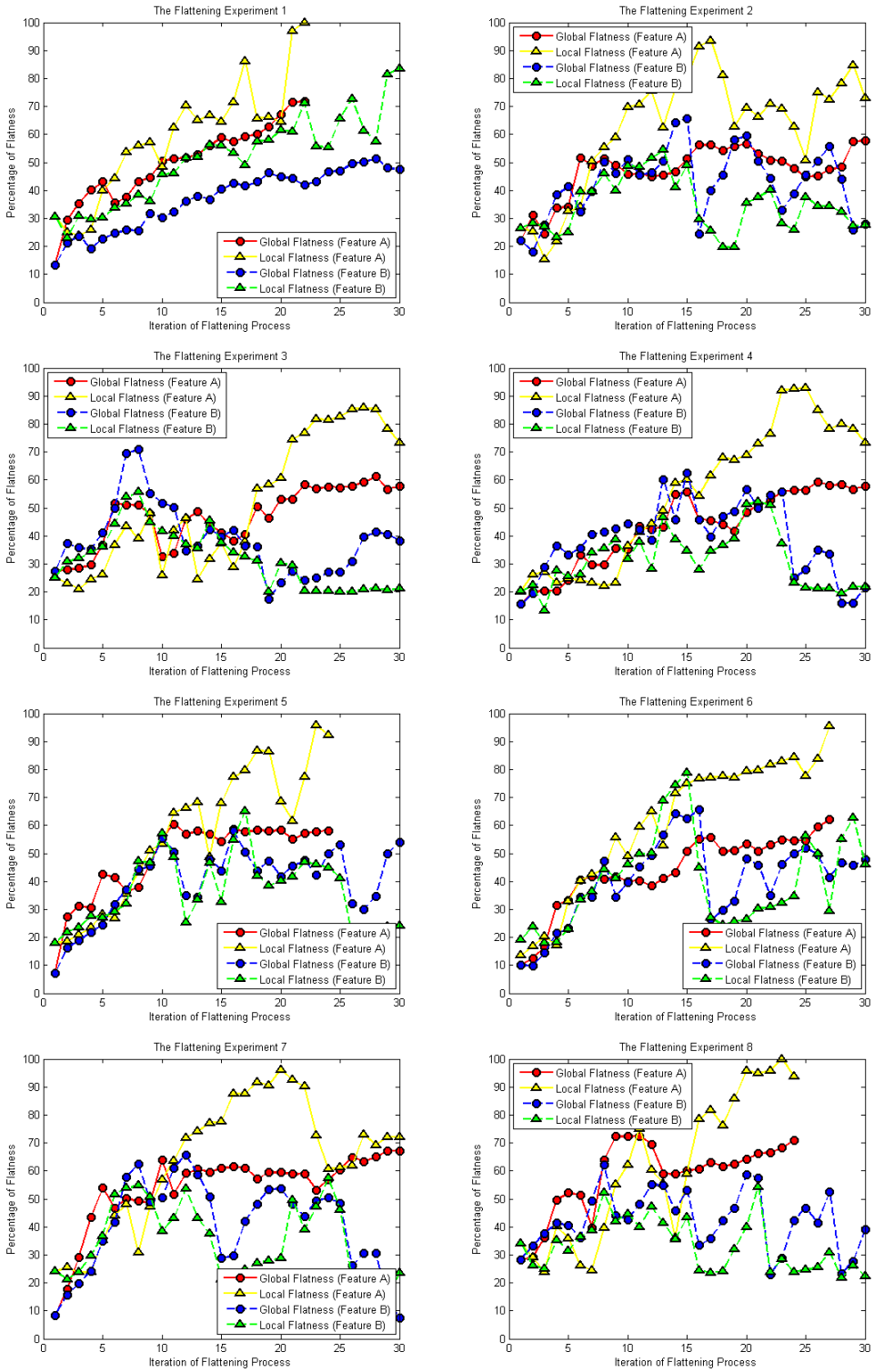


Figure 3.11: Comparison of flattening efficiency between two proposed features. Here feature A refers to geometry-based feature and B clustering-based feature. Red and blue lines show the global flatness of features A and B, respectively. Yellow and green lines illustrate the local flatness of features A and B, respectively. In these experiments, the 'flatness' scores are recorded at each iteration and the flattening is repeated till no wrinkle can be detected.

indexes are linearly normalised to lie in the range from 100% to 0%, depending on the their minimum and maximum possible values[3]. Specifically, the lines in Figure 3.11 with circular markers denote the change in 'global flatness', lines with triangular markers indicate the change in 'local flatness', solid lines indicate the 'flatness' achieved by the geometry-based feature and the dashed lines indicate the 'flatness' achieved by the clustering-based feature.

Among these 8 experiments, it can be observed that, for both features, the 'global flatness' scores correlate positively with the 'local flatness' scores during the flattening process. Even though both flatness scores exhibit different numeric ranges, the temporal correlation between these scores would be expected to be observed if flattening was progressing for both scoring methods. The local flatness curves typically fluctuate more than the global flatness curves since the configuration of local wrinkles is not preserved after each iteration, as observed in Figure 3.11.

By closely inspecting the plots in Figure 3.11, it is noticeable that the flatness in the geometry-based approach increases more rapidly than that in clustering-based approach (especially in experiments 1, 2, 5, 6, 7 and 8) for the first 15 iterations. For 'local flatness', in most of the experiments (1, 3, 5, 6 and 8), the geometry-based approach (yellow curve) exhibits a rapid increase, while the clustering-based approach (green curve) usually fluctuates around 40% after approaching its best 'local flatness' score.

To explain the behaviour behind these curves, representative experiments are selected for a further analysis. In experiment 1, both feature extraction methods exhibit similar tendencies, but the geometry-based approach is always higher than the clustering-based approach (the red curve is always higher than the blue curve; the yellow curve is always higher than the green curve). This result can be attributed to an improved description and representation encoded in the geometry-based approach. In experiments 2, 5, 6 and 8, after approaching a relatively high degree of flatness (60% 'global flatness' and 50% 'local flatness'), the clustering-based method demonstrates high variability (in the blue and green curves), which suggests that the clustering-based approach is not effective at measuring and flattening small wrinkles. In experiments 3, 4 and 7, the flatness (both 'global' and 'local') of clustering-based methods decreases (blue and green curves) after reaching maximum values. This suggests that the forces being applied to the cloth are no longer appropriate and are increasing, rather than reducing, the number of wrinkles on the cloth.

3.7.3 Evaluation and Comparison

Now a more objective statistical analysis of the experiments in Figure 3.11 is explored. The *efficiency*, *quality* and *stability* of the cloth flattening task are investigated. More specifically,

[3] As it is mentioned in the definition, the 'flatness' is inversely related to the wrinkledness values. Hence the minimal wrinkledness value corresponds to 100% flatness, and vice versa.

Table 3.1: The Required Number of Iterations (RNIs) in 8 flattening experiments.

RNI of Experiments	exp1	exp2	exp3	exp4	exp5	exp6	exp7	exp8	average	SD
Clustering-Based Method (Global)	27	8	7	**13**	10	**13**	7	8	11.625	6.675
Geometry-Based Method (Global)	**11**	**6**	**6**	14	10	15	**5**	**5**	**9**	**4.071**
Clustering-Based Method (Local)	12	12	**7**	20	10	12	**6**	**8**	10.875	4.390
Geometry-Based Method (Local)	**7**	**7**	18	**14**	**9**	**9**	10	9	**10.375**	**3.777**

the 'efficiency' is defined as how quickly the flattening strategy, using either feature type, can flatten the cloth to a relatively high degree of flatness. The 'quality' is defined as the highest flatness value achieved at the end of the experiment, and the 'stability' to be how *consistent* the flattening approach is (i.e. how reliably the feature extraction approach can perform the flattening task).

To compare 'efficiency', the required numbers of iterations (RNI) to reach 50% of global and local flatness are observed. As shown in Table 3.1, the RNI of the clustering-based feature for 'global flatness' is equal to, or higher than, the geometry-based approach in 6 out of 8 experiments. The average RNI of the geometry-based approach is 9.0, which is approximately 30% less than of the clustering-based approach (11.625). Likewise, the standard deviation (SD) of the 8 experiments using the geometry-based approach (4.071) is also lower than the clustering-based approach (6.675). This indicates that the variance of the former method is smaller over the 8 experiments. The comparison of the performance demonstrates that the geometry-based approach can perform more efficiently than the clustering-based approach for cloth flattening.

In order to describe the 'quality' of the task, Fig. 3.12 shows the highest flatness score achieved in each experiment. In the figure, the x-axis corresponds to the 'global flatness' and the y-axis corresponds to the 'local flatness'. Ideally, points should be in the upper right corner. Close inspection of this figure reveals that the 'global flatness' of the two feature approaches are very similar: the average 'global flatness' of the geometry-based approach (64.03%) is slightly higher than for the clustering-based approach (62.72%), as the 'global flatness' becomes less distinct when the cloth is relatively flat. Whereas for 'local flatness', the flatness average achieved by the geometry-based approach (94.92%) is much higher than that scored by the clustering-based approach (62.69%). This is because the geometry-based approach is more effective in flattening small wrinkles due to its accurate spatial location and size quantification. This analysis demonstrates that the geometry-based approach delivers a greater degree of local flatness than the clustering-based approach.

Finally the 'stability' of the two flattening methods is investigated by computing the stan-

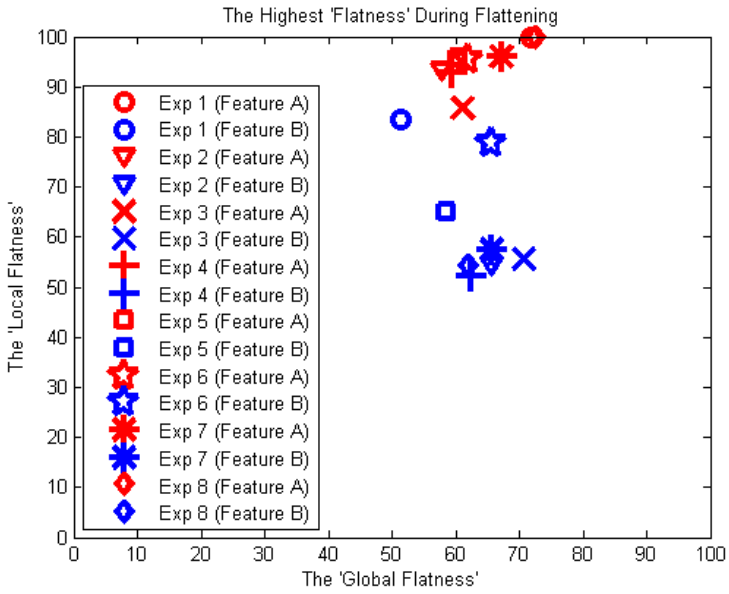


Figure 3.12: Comparison of flattening quality between two proposed features. Here feature A refers to geometry-based feature and B clustering-based feature. In this figure, the highest global and local flatness values are shown from the whole flattening process. The red markers show the approach using the geometric-based feature and blue markers, the approach using the clustering-based feature.

Table 3.2: Standard deviations of differences in 'global flatness' between iterations in 8 flattening experiments. The lower SD are in boldface.

SD of Difference	exp1	exp2	exp3	exp4	exp5	exp6	exp7	exp8	average	SD
Clustering-Based Method	**2.67**	11.54	7.78	10.40	8.29	10.04	9.64	12.10	9.10	2.965
Geometry-Based Method	4.26	**4.97**	**5.67**	**4.18**	**5.71**	**3.87**	**6.07**	**7.10**	**5.23**	**1.108**

dard deviation (SD) of the differences in flatness at each iteration. Due to the instability in the local flatness (as discussed above), only global flatness is considered in this part of experiments. The results presented in Table 3.2 indicate that the flattening process of the geometry-based approach has a lower SD than the clustering-based approach in 7 out of 8 experiments. The average SD of the geometry-based method (5.23) is 42.5% lower than of the clustering-based approach (9.10). This result demonstrates that the geometry-based approach is more stable than the clustering-based approach.

Overall, by comparing the results of the three experiments described above, the conclusion is: The geometry-based approach outperforms the clustering-based approach in all three measurements (i.e. efficiency, quality and stability) investigated in this research.

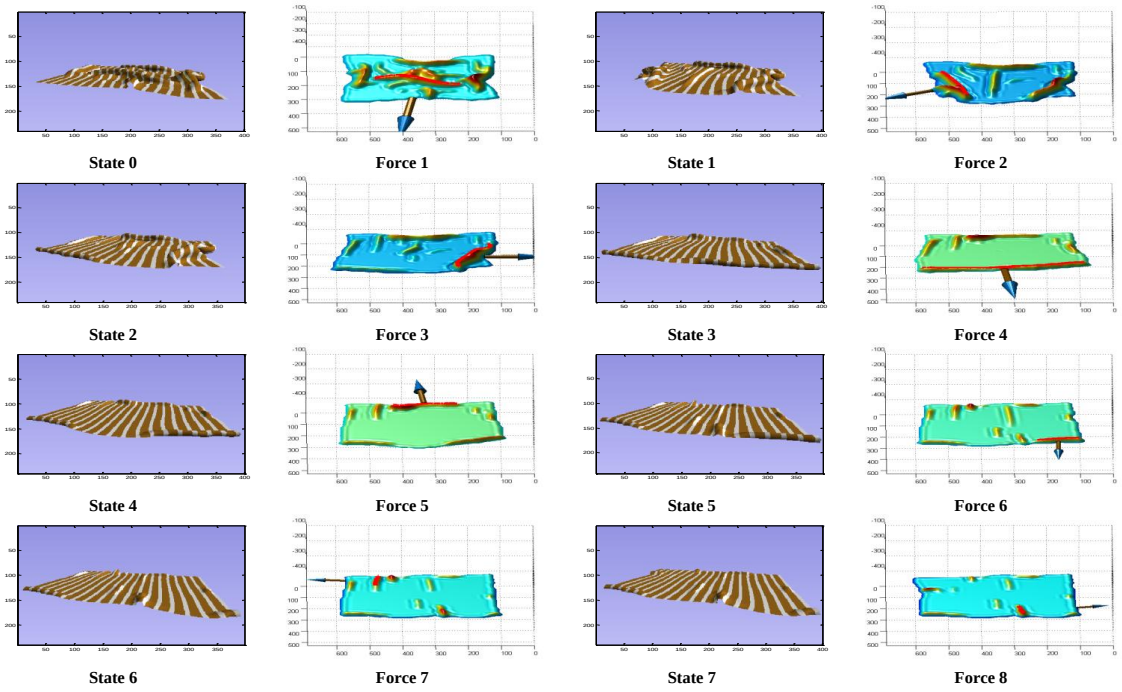


Figure 3.13: The demonstration of flattening virtual cloth using geometry-based feature. In this figure, seven iterations are shown, where the arrows represent the flattening force, and its size is positively related to the force's magnitude.

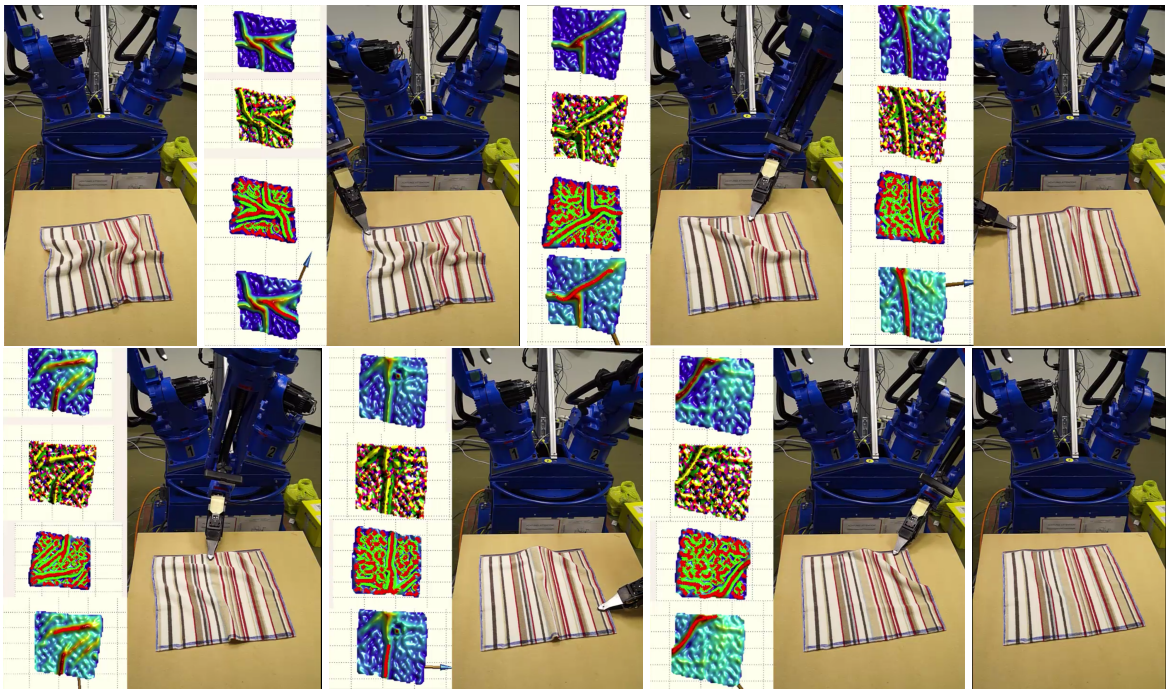


Figure 3.14: The autonomous single-arm flattening demo with 7 iterations. The video of the demo is available at https://www.youtube.com/watch?v=iOEto5Gy6vg

Table 3.3: The Required Number of Iterations (RNI) of single-arm flattening in the real robot scenario. The halting criteria is using 'local flatness'.

RNI of Experiments	exp1	exp2	exp3	exp4	exp5	exp6	exp7	exp8	average	SD
Clustering-Based Method	12	15	13	16	12	10	11	14	12.88	2.96
Geometry-Based Method	10	12	7	6	8	10	11	7	**8.88**	**1.11**

3.7.4 Validation in Robot Testbed

Additionally, this proposed method is integrated into an autonomous flattening pipeline using CloPeMa robot in order to evaluate the real function of the proposed virtual clothes manipulation system. In this experiment, eight single-arm flattening experiments are conducted for each feature representation method, and the results are presented in Table 3.3. To be specific, depth data is acquired by ASUS Xtion Pro and 50% 'global flatness' is used as the halting criterion. The results of real robot experiments conforms with those obtained from simulated experiments. The average RNI of geometry-based feature for completing a flattening experiment is 8.8, which advances the performance of clustering-based feature (average RNI of 12.88). Although it is impossible to duplicate the same cloth configurations for comparison in real scenario, this result underpins the reality of simulated flattening experiments from the statistical perspectives.

3.8 Conclusion

This chapter proposed an advanced geometry-based clothes configuration analysis approach for visually-guided clothes manipulation, which is the preliminary version of the proposed visual architecture. This approach is able to parse 3D clothing configurations by the means of surface shape and topology analysis, thereby detecting and quantifying high-level wrinkle configurations. Furthermore, this approach is capable of sufficient parametrising and accurate mapping of these wrinkle features to allow them to be used in robotic flattening tasks involving the manipulation of garments and flexible materials.

The comparison between geometry-based features and the baseline clustering-based features, i.e. flattening simulated wrinkled cloth of the same initial configurations using the same flattening strategy, indicates that the geometry-based approach outperforms the preliminary statistical clustering-based approach with respect to the flattening performance. Based on the evaluation of flattening *efficiency*, *quality* and *stability*, the comparison results demonstrate that an improved cloth flattening performance can be achieved by means of advanced visual perception of the key wrinkle structures in terms of their detection, localisation, parametrisation and mapping. In turn, an improved wrinkle map facilitates the inference of the orien-

tation and magnitude of manipulation action and thereby improves the performance of the flattening strategy.

The experimental results evaluated in the physical clothes simulation verify one of the hypotheses of this thesis:

- In order to manipulate a garment, it is necessary to understand the garment's local surface shapes in order to identify 3D garments structures for the grasping or flattening purposes. In addition, metric information specify the dimensions of these structures must also be recovered through vision in order to indicate the pose and motion of the end effector used to manipulate these structures.

The proposed two perception methods are devised to parse the garment's structures through local shape analysis – one is from discontinuity analysis and the other is from the shape and topology analysis. The experimental results demonstrate that they are able to provide the metric information of the wrinkles in different degree of accuracy, and tend to guide the simulated robot to conduct the flattening action. Moreover, the perception method of a greater degree of accuracy can accelerate the flattening process.

In the next chapter, this proposed geometry-based visual perception will be refined and extended to a more generic visual perception architecture, which is capable of guiding other laundering tasks (i.e. grasping) and more complex flattening tasks (e.g. dual-arm flattening of t-shirts, pants, etc.). Moreover, in chapter 4, the visual architecture is integrated into dual-arm CloPeMa robot and the performance of visually-guided manipulations are evaluated in the real robot testbed.

Chapter 4

Clothes Manipulation Intergated with Dual-arm Robot

In this chapter, the preliminary geometry-based feature reported in Chapter 3 is refined and extended to a more generic visual perception architecture which is able to guide the dual-arm robot to conduct multiple clothes manipulation tasks (i.e. grasping and flattening garment). The proposed visual perception architecture is a three-layer hierarchical structure, evolving the visual representation from meaningless curvatures to semantic descriptors. This visual perception architecture is integrated into the dual-arm CloPeMa robot system incorporating off-line hand-eye calibration, RGB-D sensing, segmentation, manipulation strategy, motion planing and visual/tactile feedbacks. Differing from the evaluation in Chapter 3, in this chapter the manipulation performances are evaluated from the statistics collected through sufficient repetition of experiments in real robot scenario (as shown in Fig. 4.1).

4.1 Introduction

This chapter presents the 'parsing' part of the proposed visual architecture which parses the 3D clothes configurations hierarchically from low-level curvatures, across mid-level geometrical shape & topology analysis, and finally approaching high-level semantic surface structure descriptions i.e. grasping triplets and wrinkles. This 'parsing' approach is integrated into the autonomous robot manipulation system conducting autonomous grasping and dual-arm flattening. The system starts with image capturing, then goes through stereo-matching and 3D reconstruction, delivering the RGB-D data, then a supervised grab-cut segmentation is applied to obtain the segmentation of the garment. Then the raw 3D garment data is hierarchically parsed by the proposed visual perception architecture to obtain the grasping/flattening primitives – 'triples' and 'wrinkles'. The flatness and volume are used to

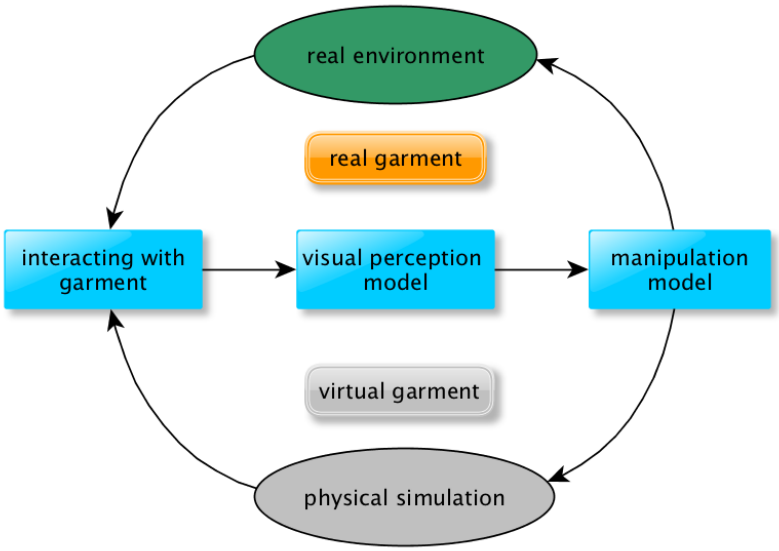


Figure 4.1: The highlighted framework of visually-guided clothes manipulation.

quantify grasping triples and wrinkles respectively, and thereafter, triplets and wrinkles are ranked by a greedy strategy that can identify the best grasping triplet and the largest wrinkle present. The trajectory of motion is optimised for grasping a specific triplet or flattening a wrinkle is formulated based on dual-arm manipulation. The proposed autonomous grasping is evaluated in both single-trial and interactive-trial experiments, showing robustness among the clothes types. And the validation of the reported autonomous flattening behaviours has been undertaken and has demonstrated that dual-arm flattening requires significantly fewer manipulation iterations than single-arm flattening. The experimental results also indicate that the dexterous clothes operation (such as flattening) is significantly influenced by the quality of the RGB-D sensor − using a customized off-the-shelf high-resolution stereo-head outperforms the commercial low-resolution Kinect-like cameras in terms of required number of flattening iterations (RNIs).

This chapter is structured as follows: Section 4.2 gives the motivation and objectives of this chapter. Section 4.3 gives a overall schema of the proposed visual perception architecture and integrated autonomous pipeline. In Section 4.4, the hierarchical visual architecture for generic garment surface analysis is introduced. Section 4.5 presents the proposed visually-guided grasping approach and dual-arm flattening approach, which are on the basis of proposed visual perception architecture. The experimental validations of the proposed autonomous grasping and flattening are detailed in Section 4.6. The conclusion of this work is given in Section 4.7.

4.2 Motivation and Objectives

Dexterous manipulation of clothing has demonstrated to be a difficult task for autonomous robotic systems because it requires precise parsing of the deformable configuration of garments. Form the literature review about visually-guided clothes manipulation (reported in Section 2.6.3), Kinect-like cameras are widely-used for clothes perception, and various manipulation approaches are proposed for resolving clothes gasping, unfolding and folding problems. Kinect-like cameras can provide a cheap and real-time depth sensing with accuracy of approximately 1cm. Under such accuracy, the Kinect-like cameras can barely capture the small landmarks or estimate the magnitude of bending of clothes surfaces accurately, which are required for dexterous manipulations such as grasping and flattening. Moreover, these predominant reported methods are constrained to a specific garment or task at hand. In other words, the state-of-the-art visual perception approaches for clothes manipulation are not generic enough for more than one tasks. This can be attributed to the lack of sufficient parsing of clothes configuration, by which the generic landmarks can be localised and parametrised and tend to indicate dexterous manipulation. To the best of the author's knowledge, Ramisa et al. [Ramisa et al., 2013] proposed a 3D descriptor that is exploited in clothes grasping, wrinkle detection, and category recognition tasks, which is the only generic approach for multiple clothes perception tasks. Their proposed approach simply adapts all tasks into a *black-box classification* problem rather than parse the clothes appearances into details. Moreover, they evaluate their manipulation performance in annotated datasets as opposed to real-life experiments. The calibration and integration error and the offset and mistake caused by labelling are not considered.

From the arguments above, the state-of-the-art visually-guided dexterous clothes manipulation have the following limitations: Firstly, Kinect-like and low-resolution depth cameras are not precise enough to sense garment details and as a consequence dexterous visually-guided manipulations are extremely difficult to be guided by these cameras. These types of cameras therefore limit the application scope and capabilities of robots. Secondly, existing approaches for visually-guided clothes manipulation usually focus on specific tasks rather than a generic parsing of the garments' geometrical configuration with sufficient understanding of surface shapes and topologies. As a consequence, most of the existing approaches are unlikely to be extended to multiple laundering tasks.

In order to offset these limitations of the predominant reported approaches, the objectives of this chapter are two-fold: First, adapting an off-the-shelf, high-quality active binocular robot head for clothes perception instead of using Kinect-like cameras, which outperforms the depth sensing in both resolution and quality. In this work, the binocular robot head incorporating a GPU stereo matcher tuned specifically for clothing provides accurate depth sensing for parsing the garment's 3D configuration. Second, solving garment perception and

manipulation tasks on the basis of sufficient parsing of the 3D configuration of the garment by the means of surface shape and topology analysis. Thus, a bottom-to-top visual architecture (the 'parsing' part) is proposed to achieve a full understanding of the 3D garment's configuration. The proposed architecture parses 3D garment surfaces from low-level curvatures, via mid-level shapes and topologies, finally to high-level semantic structures that can be adapted to multiple manipulation tasks. Consequently, through parsing the garment's configuration, the precise grasping and flattening manipulating skills can be indicated from visual guidances.

4.3 An Overall Schema

4.3.1 The Hierarchical Visual Architecture

In this work, a hierarchical visual perception architecture is proposed, which is able to be adapted to a range of clothes manipulation tasks. This architecture is based on 3D surface analysis including three layers of visual features: low-level curvature features, mid-level shape/topology features, and high-level semantic features for specific tasks. As shown in Fig. 4.2, the feature architecture is orientated from the surface curvatures (low-level). Then the geometry-based mid-level surface features, i.e. surface shapes and surface topologies (ridges and wrinkle's contours), can be estimated by the maximal and minimal curvatures. In order to adapt the surface shape and topology knowledge to garment manipulation tasks, high-level semantic features, i.e. grasping triplet (the atom of grasping candidates) and wrinkle descriptor, are proposed.

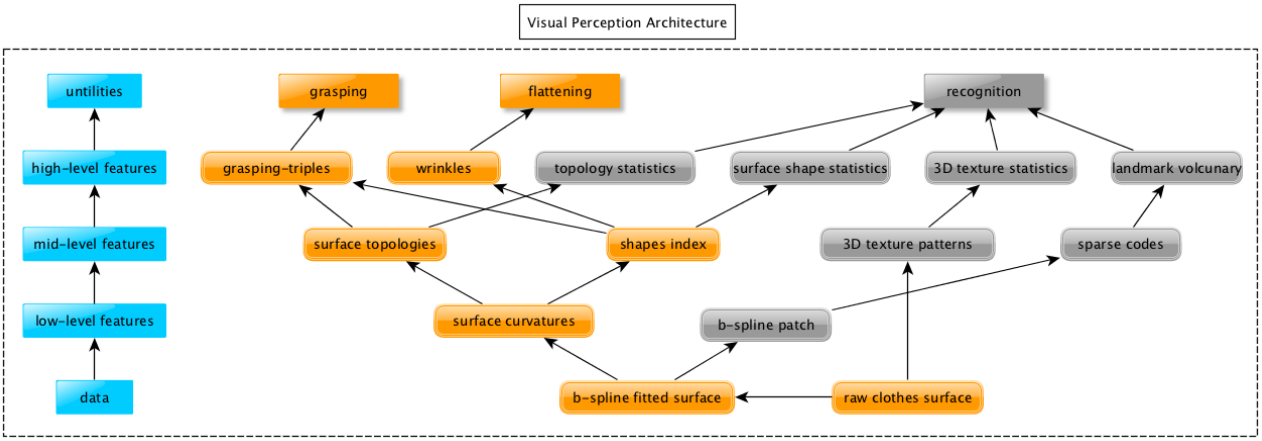


Figure 4.2: The hierarchical visual architecture for visually-guided clothes manipulation.

4.3.2 The Pipeline of the Integrated Autonomous Systems

The whole integrated pipeline for conducting autonomous grasping and flattening is shown in Fig. 4.3. There exist four stages for this integrated pipeline: (0) off-line calibration, (1) stereo-matching and 3D reconstruction, (2) hierarchical visual parsing, and (3) grasping/flattening motion planning.

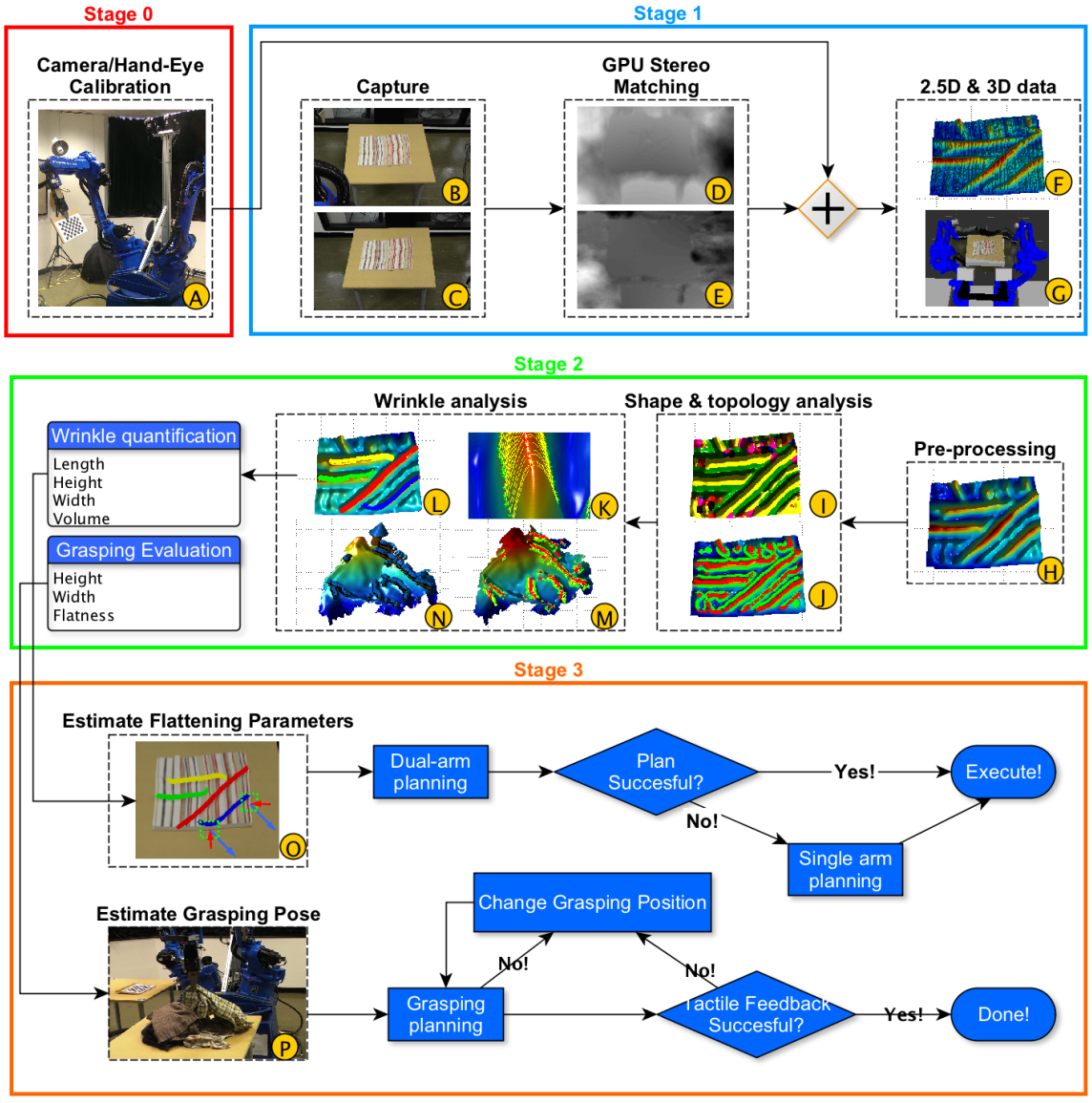


Figure 4.3: The whole pipeline for autonomous grasping and flattening.

In stage 0, the PTUs are fixed in the original positions, and the calibration target (chess board) grasped by the gripper is explored in the visible scope. During this procedure, images are captured by the static stereo cameras and the intrinsic parameters of the stereo cameras are estimated by OpenCV calibration routines[1]. Simultaneously, the transform between cameras

[1]http://opencv.org

and robot gripper is estimated by Tsais hand-eye calibration routines [Tsai and Lenz, 1988, 1989], thereby linking the stereo cameras into the kinematic chain of the robot.

Having calibrated and integrated the stereo-head, the next stage is stereo-matching and 3D reconstruction. In this procedure, a pair of images are captured simultaneously by the left and right cameras. The C3D matcher [Siebert and Urquhart, 1995, Zhengping, 1988] is employed to find the horizontal and vertical disparities of the two images. In the implementation of CloPeMa head, C3D matcher is accelerated by CUDA[2] GPU paralleling programming [Cockshott et al., 2012a] to produce a 16 mega-pixel depth map in 0.2 fps. A GMM-based grab-cut [Stria et al., 2014a] is employed to detect and segment the garment from the RGB image and then mask the point cloud.

The next stage is the proposed hierarchical 3D clothing surface parsing for detecting and parametrising the clothing's landmarks. The raw depth map of the clothing surface are approximated to obtain a geometry continuous surface. Following the proposed visual architecture, low-level curvature features, mid-level shape and topology features, and high-level landmark features are extracted from the depth map hierarchically.

The final stage is the visually-guided manipulation. The semantic structures such as grasping triples and wrinkles are localised and parametrised in the last stage, from which the poses of grippers can be indicated. *Moveit* library[3] is used to plan the trajectories of grasping and flattening poses. For dual-arm flattening, a collaborative pose selection mechanism is proposed in order to enhance the success rate of the dual-arm planning.

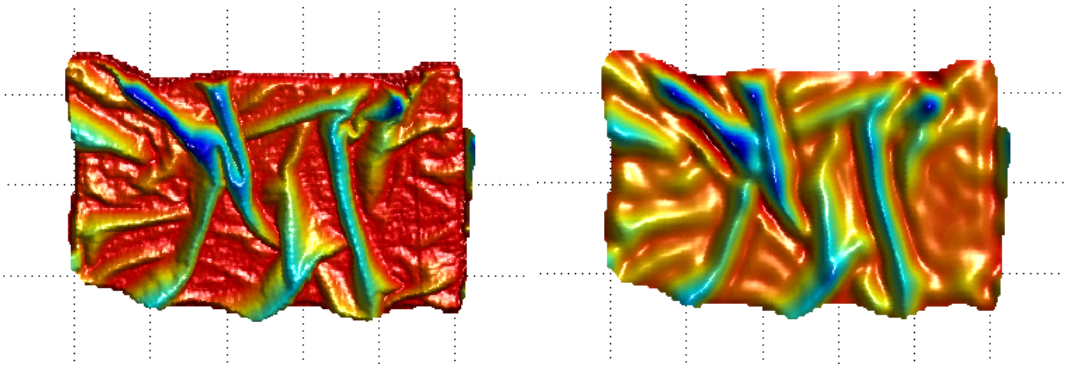
4.4 Hierarchical Visual Architecture

In this section, a piece-wise B-Spline surface fitting is adapted as pre-processing in Section 4.4.1, and the low-level feature extraction is presented in Section 4.4.2. In Section 4.4.3, surface shapes and topologies are introduced as the mid-level features. Afterwards, two high-level features i.e grasping triplets and wrinkle description, are reported in Section 4.4.5 and Section 4.4.5.

4.4.1 Pre-Processing: B-Spline Surface Fitting

As geometry-based range features such as curvatures and shape index are extremely sensitive to high frequency noise, a piece-wise B-Spline surface approximation is used to fit a continuous implicit surface onto the original depth map. An example of the fitted surface is shown in Fig.4.4. More details are presented in Appendix A.

[2]https://developer.nvidia.com/cuda-zone
[3]http://moveit.ros.org/



(a) The original range map produced by CloPeMa stereo robot-head. This example is a shower towel.

(b) The B-spline fitted surface with C1 continuity.

Figure 4.4: An example of the proposed piece-wise B-Spline surface fitting.

4.4.2 Low-Level Feature: Surface Curvatures Estimation

To compute curvatures from depth, 2.5D points in the depth map (i.e. x, y and depth $-$ x and y are in pixels while depth is in metres) are examined pixel by pixel in order to find if they are the positive extrema along the maximal curvature direction. That is, given a depth map I, for each point p in I, the mean curvature C_m^p and Gaussian curvature C_g^p are firstly calculated by Eq. 3.2 and Eq. 3.3, where first derivatives f_x^p, f_y^p, and second derivatives f_{xx}^p, f_{yy}^p,f_{xy}^p are estimated from the B-Spline control points and the corresponding derivatives of base functions (more details are shown in Appendix A). Then, the maximal curvature k_{max} and minimal curvature k_{min} can be calculated by C_m^p and C_g^p (shown in Eq. 3.4).

4.4.3 Mid-Level Features: Surface Shapes and Topologies

Surface Shape Analysis

Shape index [Koenderink and van Doorn, 1992], performs a continuous classification of the local shape within a surface regions into real-value index values, in the range [-1,1]. Given a shape index map S, the shape index value S^p of point p can be calculated by Eq. 3.6 [Koenderink and van Doorn, 1992]: where k_{min}^p, k_{max}^p are the minimal and maximal curvatures at point p computed using Eq.3.4. The shape index value is quantised into nine uniform intervals corresponding to nine surface types – *cup*, *trough*, *rut*, *saddle rut*, *saddle*, *saddle ridge*, *ridge*, *dome* and *cap*.

In order to parse the shape information exhibited by the visible cloth surface, the shape index map is calculated from each pixel of the depth map and a majority rank filtering is applied. This non-linear filtering removes outlier surface classifications and can be tuned to produce a relatively clean classification of shape types over the cloth surface. An example can be

seen in Fig. 4.3-I. It is worth noting that, the shape types 'rut' and 'dome' can be used to recognise the junction of multiple wrinkles thereby splitting wrinkles (as shown in Fig. 4.5).

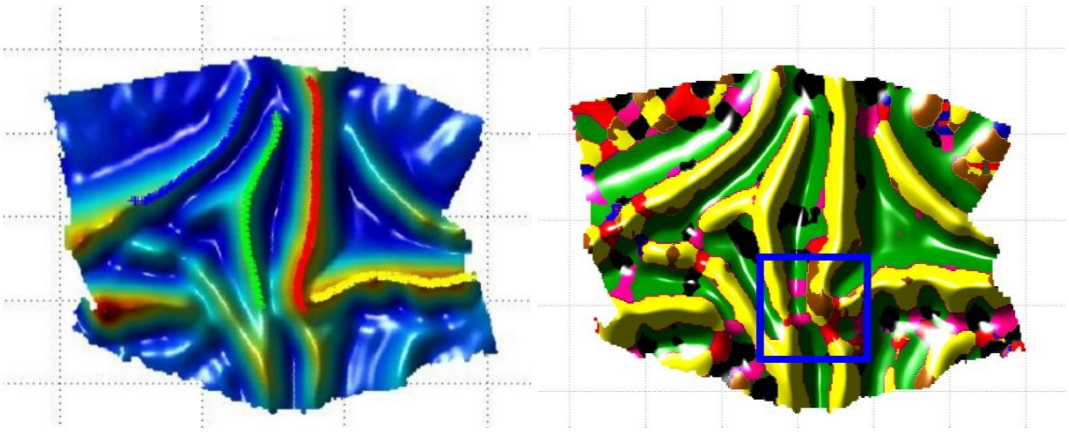


Figure 4.5: An example of splitting wrinkle using Shape Index. In highly wrinkled situations, the shape of wrinkles at junctions are classified as dome or rut (as shown in brown and red colours); this classification is used to separate jointed wrinkles in this work.

Surface Topologies Analysis

Among all the shape types, *ridges* is of critical importance in the analysis and description of wrinkles. In this chapter, the definition of *ridges* shares similarities to that given by Ohtake et al. [Belyaev and Anoshkina, 2005]. The main difference is that instead of estimating curvatures from a polygon mesh, surface curvatures are calculated using differential geometry, obtained directly from the depth map (as it is presented in Section 4.4.2).

As it is illustrated in Section 3.5, *surface ridges* are therefore the positive extrema of maximal curvature [Ohtake et al., 2004] while *the wrinkle's contour* is the boundary of the concave and convex surfaces of the garment.

From the nine shape types, four are convex (i.e. saddle ridge, ridge, dome, cap) and the rest are concave (i.e. cup, trough, rut, saddle rut, saddle). Thereby, the wrinkle's contour can be estimated. Alternatively, the boundary of the convex and concave surface can be more robustly estimated by computing the zero-crossing of the second derivatives of the garment's surface. For the implementation of this work, a Laplace template window of size 16×16 is applied on the depth map in order to calculate the second order derivative. After the wrinkle's contour has been detected, the garment surface topologies are fully parsed. An example can be seen in Fig. 4.3-J.

4.4.4 High-Level Features - Grasping Triplets

As illustrated in Section 3.5, a wrinkle comprises a continuous ridge line localised within in a region where the surface shape type is 'ridge'. The wrinkle is delimited (bounded) by two contour lines, each located on either side of the maximal curvature direction. A wrinkle can be quantified by means of a triplet comprising a ridge point and the two wrinkle contour points located on either side of the ridge, along the maximal curvature direction (as shown in Fig. 3.5).

The above definition is inspired by classical geometric approaches for parsing 2.5D surface shapes and topologies (i.e. shape index [Koenderink and van Doorn, 1992], surface ridges and wrinkle's contour lines). In this work, the height and width of a wrinkle are measured in terms of triplets. Accordingly, triplets can also be used as the atomic structures for finding and selecting grasping points (shown in Fig. 4.3-N).

Triplets Matching

From wrinkle's geometric definition, the maximal curvature direction θ can be calculated by Eq. 3.5. Given a ridge point p_r in a depth map I with scale φ_{L1}, this proposed method searches for the two corresponding contour points (p_c^l and p_c^r) over the two directions defined by θ and its symmetric direction using a depth based gravity-decent strategy. If the searched path is traversed in the same 'ridge' region as p_r (shown as yellow in Fig. 4.3-I), the process will continue. Otherwise, the searching will be terminated. Algorithmic details of triplet matching are described in Algorithm 5. This algorithm presents the process of matching the left contour point, however, the same algorithm holds true while searching for the right contour point.

Theoretically, every ridge point should be matched with its two corresponding wrinkle contour points. Due to occlusions and depth sensing errors, some wrinkle points fail to find their associated contour points and therefore do not generate a triplet. In order to eliminate the uncertainties caused by occlusions and errors, only triplets whose ridge points matched with both two wrinkle contour points are regarded as valid primitives for wrinkle quantification. An example of triplets matching is shown in Fig. 4.3-K and M. Given a triplet t_p containing one ridge point p_r and two wrinkle contour points p_c^1 and p_c^2, the height h_t and width w_t can be calculated from the embedded triangle (triplets) using Eq. 3.8. It is worth noting that, the triplet's points are transformed to the world coordinates, and as a consequence the unit of height h_t and width w_t is in meter.

4.4.5 High-Level Features: Wrinkle Description

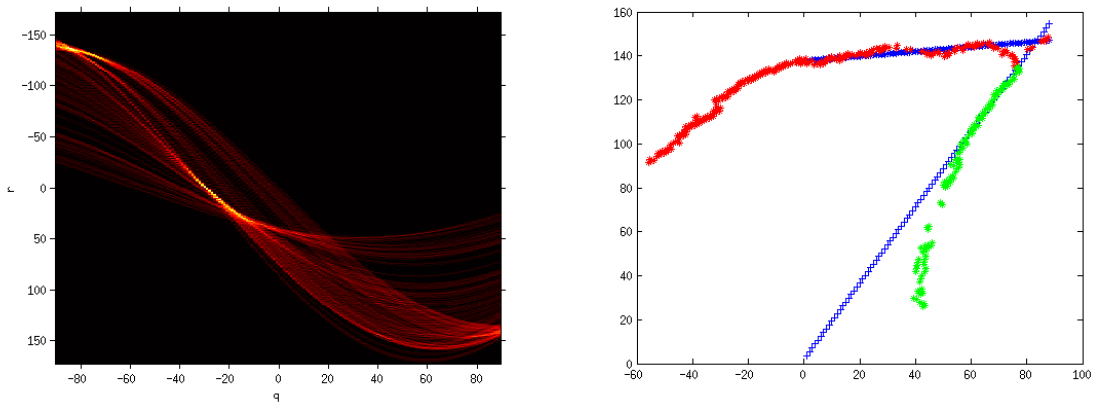
Wrinkle Detection

The wrinkle detection consists of two steps: first, connecting ridge points into contiguous segments; second, grouping found segments into wrinkles (Fig. 4.3-L). More details are given in Section 3.5.

After wrinkles have been detected, for each wrinkle, a fifth order polynomial curve is fitted along its ridge points. A high order polynomial curve is adopted in order to ensure that it has sufficient flexibility to meet the configuration of the wrinkles (fifth order works well in practice). The polynomial curve denotes the parametric description of a wrinkle, and the curve function is defined as:

$$f(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + f, \tag{4.1}$$

Hough Transform-Based Wrinkle Splitting



(a) The Hough line parameter space. Here, x-axis refers θ and y-axis refers ρ.

(b) The joined wrinkle points are split by two main directions.

Figure 4.6: Splitting joined wrinkles through Hough-Transform based wrinkle direction analysis. In Fig. 4.6(a), the two peak points refer to the two main directions of the joined wrinkles. In Fig. 4.6(b), the two main hough line directions are plotted as blue line, and the points of joined wrinkle are split corresponding to these two direction, shown as red and green respectively.

As reported in Section 4.4.3, *Shape Index* is used to find junctions of wrinkles, where the shape types 'rut' and 'dome' are used as the visual cues for splitting wrinkles (as shown in Fig. 4.5). Furthermore, an additional Hough transform-based wrinkle splitting approach is proposed. In our approach, the joined wrinkles are parametrised as straight lines in Hough space in order to find the primary directions of the joined wrinkles. Specifically, the Hough

transform-based wrinkle splitting is employed if the quality of wrinkle fitting (the *RMSE* of the polynomial curve fitting above) is not acceptable. That is, each 2D point on the fitted wrinkle is projected as a curve in the Hough line parameter space. Afterwards, peaks in the Hough space are ranked and the two largest peaks indicate the two main directions of the joined wrinkles. In order to avoid choosing two peaks originating from the same wrinkle, the two largest peaks should satisfy a non-locality constraint. In the implementation of this work, the value of 20 degrees works well in practice. Then, wrinkle points can be split into two subsets depending on the two largest peaks. An example of this proposed splitting is shown in Fig. 4.6. Finally, new polynomial curves are approximated on the split points respectively. This splitting procedure will be performed recursively until all the wrinkles are below an optimal *RMSE* value (in practice, a value of 2 pixels works best for the implemented vision system of this work). Algorithm 6 details the proposed Hough Transform based wrinkle splitting approach.

Algorithm 6 The Hough Transform based wrinkle splitting approach.

In: The detected wrinkles' points for splitting $\{P_x, P_y\}$, the threshold tolerance, tol, of the *RMSE* wrinkle fitting, and non-locality constraints threshold, $thres_\theta$.

Out: The splited wrinkles' points $\{P_x^1, P_y^1\}$ and $\{P_x^2, P_y^2\}$.

Approximate polynomial curve to $\{P_x, P_y\}$, and calculate the fitting error $rmse$.
if $rmse$ is larger than tol **then**
    Transform $\{P_x, P_y\}$ to Hough space, and get ρ and θ
    Find the peak points in hough space and rank them w.r.t the accumulator values $\{\hat{p}_1, ..., \hat{p}_{n_p}\}$, here $\hat{p}_i =< \rho, \theta >$.
    Find the two largest peak points $\hat{p}_{max_1}$ and $\hat{p}_{max_2}$ satisfying $\parallel \theta_{\hat{p}_{max_1}}, \theta_{\hat{p}_{max_2}} \parallel > thres_\theta$.
    Restore two straight lines l_1 and l_2 in image space w.r.t two largest peaks in Hough space.
    Split the wrinkles' points $\{P_x, P_y\}$ into two subsets (P_x^1, P_y^1 and P_x^2, P_y^2) through calculating the minimal *Hausdorff* distances to l_1 and l_2.
else
    $\{P_x^1, P_y^1\}$ = $\{P_x, P_y\}$, and $\{P_x^2, P_y^2\}$ is empty.
end if

return $\{P_x^1, P_y^1\}$ and $\{P_x^2, P_y^2\}$.

Wrinkle Quantification

Shape Index classifies surface shapes without measuring surface magnitude. This proposed method therefore measures a wrinkle's surface magnitude by means of triplets (*Definition 3*). The definition of triplets is the same as that in Section 4.4.4. Whereas, the direction of triplet matching direction θ is estimated from the parametrised wrinkle description, which is more robust than that estimated from the maximal curvature direction. To be more specific,

θ is computed from the perpendicular direction of the tangent line of the fitted curve on the observed wrinkle (i.e. fifth order polynomial curve in Eq. 4.1). The tangent direction can be calculated as:

$$\alpha = \arctan(5ax^4 + 4bx^3 + 3cx^2 + 2dx + e). \tag{4.2}$$

Given a ridge point p_r in the depth map I, both left and right directions are searched until the corresponding wrinkle's contour points p_c^l and p_c^r are found. The pseudo code of the triplet matching heuristic is described in Algorithm 5 (Section 3). The three points that define a triplet are therefore used to measure the height and width of a wrinkle. That is, given a wrinkle, ω, containing a set of triplets $\{t_1, ..., t_{N_r}\}$, the width, w_w and height, h_w are calculated as the mean value of the width and height values of ω's triplets:

$$w_\omega = \sum_{t_i \in \omega}^{N_r} w_{t_i}/N_t, \;\; h_\omega = \sum_{t_i \in \omega}^{N_r} h_{t_i}/N_t, \tag{4.3}$$

where ω_{t_i} and h_{t_i} are the width and height of the ith triplet of the wrinkle ω.

For garment flattening, the physical volume of the wrinkle is adopted as the score for ranking detected wrinkles. PCA is applied on x-y coordinates of the largest wrinkle in order to infer the two grasping points and the flattening directions for each arm. More specifically, a 2 by 2 covariance matrix can be calculated from x and y coordinates, and then the principal direction of this wrinkle can be obtained by computing the eigenvector with respect to the largest eigenvalue. To obtain the magnitude that the dual-arm robot should pull in order to remove the selected wrinkle, the geodesic distance between the two contour points of each triplet are estimated. Section 4.5.2 details how these estimated parameters are used for flattening a garment.

4.5 Autonomous Garment Manipulation

This section presents the autonomous robot clothes manipulation systems with integrated visual perception architecture. The autonomous grasping approach is reported in Section 4.5.1 and dual-arm flattening is detailed in Section 4.5.2.

4.5.1 Heuristic Garment Grasping

In this research, two visually-guided heuristic grasping strategies are proposed, in which the high-level grasping triplet feature (Section 4.4.4) is adapted as the grasping location. Both

strategies depend on an outlier removal strategy and grasping parametrisation for optimal garment manipulation as described in the following subsections.

Central Wrinkle's Points Estimation

Due to stereo matching errors caused by occlusions, inaccurate and incorrect topological descriptions may be detected, thereby affecting the estimations of grasping candidate. A central point evaluation mechanism is therefore devised to deal with isolated and inaccurate detections. This mechanism consists of computing the continuous hyper-exponential distribution of grasping triples. That is, a *Mahalanobis* distance based non-linear filtering is applied. Given a grasping triplet t_i and the size of filter window[4], its *Mahalanobis* distance can be calculated as follows:

$$D_{Mahalanobis}(p_{t_i}, p_T) = \sqrt{(p_{t_i} - \mu_T)^T \Sigma^{-1} (p_{t_i} - \mu_T)} \tag{4.4}$$

where p_{t_i} is the $x - y$ coordinate of t_i, T refers to all the triples within the filter window, μ_T is the mean of the $x - y$ coordinates of all triples, and Σ is the covariance matrix among all grasping triples within this region with respect to their spatial coordinates.

The probability of a grasping triplet being an outlier depends on the distance and direction within the hyper ellipsoid distribution. Hence, grasping triplets that are greater than a threshold[5] are treated as outliers and are removed from the list. This filtering is applied to every grasping triplet to probe whether it is an eligible grasping candidate.

Grasping Parameter Estimation

A *good grasping position* is considered as where the grasping region is most likely to fit the gripper's shape (constrained by the capabilities of the CloPeMa robot testbed) and at the same time is most unlikely to change the garment's configuration when grasped. That is, the gripper must get grip of a large region of the clothing surface in order to provide a firm grasp on the clothes. In this approach, two robotic poses are required for a successful grasping action. These are: *before-grasping* and *after-grasping* poses. The *before-grasping* pose is above the grasping point, while the *after-grasping* pose indicates the lowest position the gripper should reach without colliding with the surface of the garment. By interpolating these poses sequentially, the robot is therefore able to conduct a smooth grasping action.

The required parameters for completing the two grasping poses mentioned above comprise: the before-grasping pose of the gripper with respect to the robot's world reference frame,

[4]From practical experience, a filter window of 32×32 works well in practice.
[5]From practical experience, a threshold of 0.5 works well in practice.

the normal vector of grasping triplet and the rotation angle of the gripper with respect to the normal vector. The 3D position of gripper can be indicated by the detected grasping candidate. The grasping orientation of the gripper is set as the surface normal direction of the local region to grasp. In this research, the surface normals are robustly estimated from the third principal direction of PCA of local point cloud[Rusu and Cousins, 2011]. In order to obtain a robust estimation of the gripper rotation, the principal direction of graspable candidates within a local region is estimated and its perpendicular direction is used as the gripper rotation.

Grasping Strategies

In this chapter, two grasping strategies are proposed: a *height-priority* and a *flatness-priority* strategy. For the height-priority strategy, the grasping energy of the motion of the gripper is minimised by selecting the candidates from the highest graspable points with respect to the robot's world reference frame. While the flatness-priority strategy computes a $flatness$ score for each grasping candidate, t, that encodes the height, h_t, and the width, w_t of the wrinkle's topology (Eq. 4.5):

$$flatness(t) = \frac{h_t}{w_t} \tag{4.5}$$

The height-priority strategy is able to grasp the clothing with the smallest cost of motion energy, and as a consequence the trajectory of planing is simpler, and is easier to solve the inverse kinematic problem and avoid collisions during motion planning. However, the drawback is also obvious, as the height-priority strategy cannot grantee that the mechanical shape of the gripper fits the region to grasp properly. In contrast, the flatness-priority strategy chooses the grasping candidate with the largest flatness, which is able to select the region most likely to fit the gripper but can bring difficulties to solving the inverse kinematic problem and avoiding collision. In the implementation of this research, the height-priority strategy and the flatness-priority strategy are selected alternately until the grasping is completed.

4.5.2 Dual-Arm Garment Flattening

Flattening Heuristic

In this research, the heuristic flattening strategy adopts a greedy search approach, in which the largest wrinkle detected is eliminated in each perception-manipulation iteration. Only largest detected wrinkle is considered to be modified per iteration because the manipulation

errors accumulated into the system increases when considering a group of wrinkles with similar directions, and the likelihood of applying appropriate flattening is significantly reduced. Therefore, the largest wrinkle detection heuristic guarantees that a solution is achieved regardless of highly wrinkled configurations. Due to the difference of units between real and simulated environment, wrinkles are quantified according to their physical volume in this chapter. It is worth noting that here volume is used instead of log volume (used in Chapter 3) since this is simply to rank the wrinkles rather than inferring the force. The volume of a wrinkle w is given by:

$$volume_w = l_r * ((\sum_{t_i \in w}^{N_r} w_t \times h_t)/N_t), \tag{4.6}$$

where N_r is the number of fitted ridge points in w, N_t refers to the number of matched triplets, t_i is the ith triplet of w, w_t and h_t refers to the width and height of the triplet t_i, and l_r is the length of the wrinkle which is calculated by summing up the L^2 distances between every two nearest ridge points.

Poses of a Primitive Flattening Action

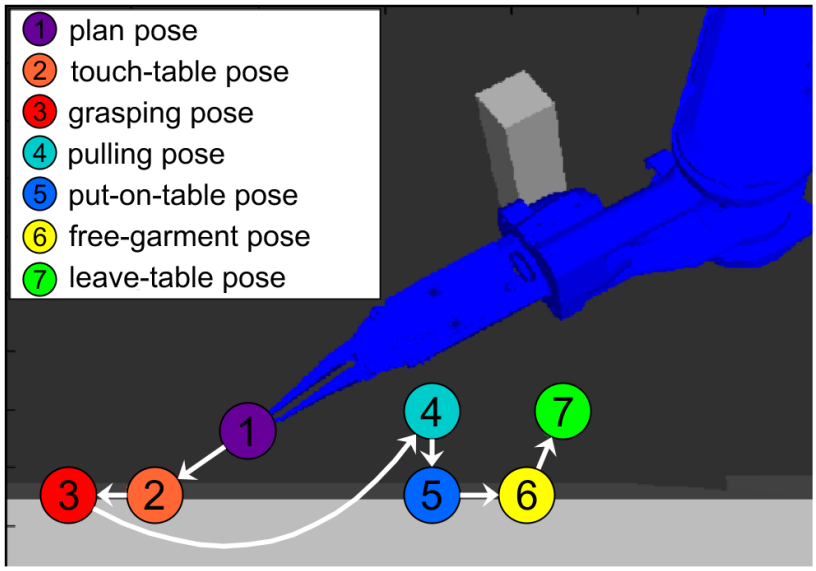


Figure 4.7: The seven poses for a robotic flattening motion. The gripper is moved to the 'plan pose', from where the trajectory of gripper is interpolated among poses sequentially in order to move the gripper. It is noticeable that the grasping direction and pulling direction are not aligned. The *plan pose*, *touch-table pose* and *grasping pose* are coplanar, while the *grasping pose*, *pulling pose*, *put-on-table pose*, *free-garment pose* and *leave-table pose* are coplanar. For the gripper state, it will be set to 'open' in *plan pose*, 'close' after *grasping pose* and 'open' again after *put-on-table pose*.

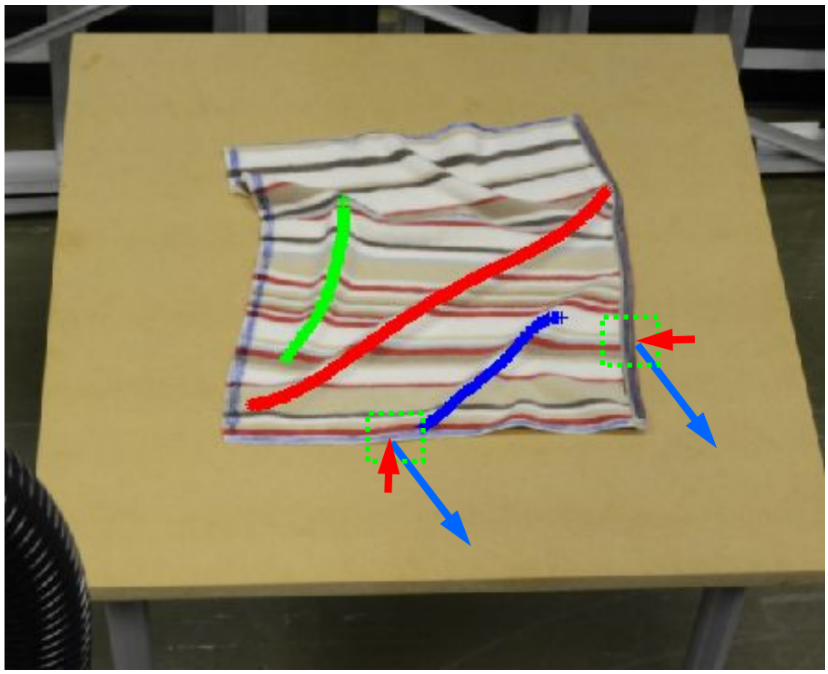


Figure 4.8: An example of detected wrinkles and the corresponding grasping poses and flattening directions of the dual-arms. The three largest wrinkles are shown, where the red one is the largest. The inferred grasping and flattening (pulling) directions are shown as red and blue arrows, respectively.

An entire flattening action consists of seven robotic poses: *plan*, *touch-table*, *grasping*, *pulling*, *put-on-table*, *free-garment* and *leave-table*. These poses are illustrated in Fig. 4.7. This figure also includes other pre-defined parameters used during the flattening task, e.g. orientation of the gripper w.r.t the table. The *plan pose* (Fig. 4.7, purple) refers to moving the robot's gripper close to the table by error-tolerance planing in preparation for flattening, then the gripper will approach the rest poses consequently by interpolating in Cartesian coordinates system. The *touch-table* and *grasping poses* (Fig. 4.7, orange and red, respectively) involve grasping the garment's boundary by interpolating the robot's motion between these two poses. The *pulling* and *put-on-table poses* pull the grasped garment according to the Geodesic distance (Eq. 4.7) and smoothly return the garment to the table. Finally, the *free-garment* and *leave-table poses* are for freeing the garment and leaving the table.

In order to calculate and interpolate these robotic poses, four parameters are required: *grasping position*, *grasping direction*, *flattening direction* and *flattening distance*. The *grasping* and *pulling poses* are estimated using these parameters. Then the other poses are inferred from the *grasping* and *pulling poses*. By interpolating these seven poses sequentially, the robot is therefore able to perform a smooth flattening action. It worth noting that for planning and interpolation, the *MoveIt* package is used[6].

[6]Available in ROS: `http://moveit.ros.org`

Flattening Parameters Estimation

Here the details about how to set the four flattening parameters (described in Section 4.5.2) are provided. As shown in Fig. 4.8, once the largest wrinkle is selected, *PCA* is employed to compute its principal direction, and the two *flattening directions* are perpendicular to the wrinkle's principal direction. After the *flattening directions* are fixed, the two corresponding intersection points on the garment contour are set as the position of the *grasping pose* (Fig. 4.7). In single arm flattening, the intersection point refers to that between wrinkle's bisector and garment's contour. While, in dual-arm flattening, wrinkles are divided into two equal segments and the two intersection points are calculated respectively. The *grasping direction* is estimated by the local contours of the *grasping positions* (as shown in Fig. 4.8). The *flattening distance* d_{w_i} of wrinkle w_i is estimated by:

$$d_{w_i} = \sum_{t_i \in w}^{N_r} (G(c_l^{t_i}, c_r^{t_i}) - E(c_l^{t_i}, c_r^{t_i}))/N_t) * Coeff_{spring}, \tag{4.7}$$

where t_i is the ith N_r triplet in w_i; $c_l^{t_i}$ and $c_r^{t_i}$ are its two wrinkle contour points; *G* refers to Geodesic distance [Sethian, 1999][7], while *E* refers to Euclidean distance. $Coeff_{spring}$ is the maximal distance constraint between particles in a mass-spring cloth model[8].

The Dual-Arm Collaboration

Because of the limitation of the robot's joints and possible collisions between the two arms, not all of the *planing poses* (the first pose of a flattening action) can be planned successfully. Therefore a greedy *pose/motion exploration strategy* is proposed (The pseudo code of the proposed algorithm is provided in Algorithm 4.5.2). This results in a significant improvement while flattening with both arms. However, if this algorithm fails, the robot only employs one arm; the arm used is selected according to the flattening direction[9].

4.6 Experiments

The experiments of this work comprise of grasping experiments shown in Section 4.6.1 and flattening experiments shown in Section 4.6.2.

[7] Gabriel Peyre's toolbox is used in the implementation of this work for calculating geodesic distance: http://www.mathworks.co.uk/matlabcentral/fileexchange/6110-toolbox-fast-marching

[8] From practical experience, the $Coeff_{spring}$ is set as 1.10 in the experiments of this work.

[9] In order to enhance the success rate of motion planing, if the flattening action is towards left, then left arm is employed; otherwise, right arm is employed.

Algorithm 7 The Pose Exploration Algorithm for Planing Dual-Arms Grasping.

In: The direction interval is d_I. The maximum numbers of exploration in each side N_E.

Out: The final planable grasping directions of two arms d_L, d_R.

Compute the ideal grasping directions D_L, D_R.
if D_L, D_R is planable **then**
  $d_L = D_L$, $d_R = D_R$;
  return d_L, d_R
end if
Set the minimal whole error of two arms $e_{min} = \infty$
for $d_l = 0; d_l \leqslant d_I \times N_E; d_l = d_l + d_I$ **do**
  for $d_r = 0; d_r \leqslant d_I \times N_E; d_r = d_r + d_I$ **do**
    Compute the error of left arm and right arm, $e_l = d_l/d_I$, $e_r = d_r/d_I$;
    Compute the whole error $e_{lr} = e_l \times e_r$;
    if d_l, d_r is planable and $e_{lr} < e_{min}$ **then**
      $d_L = d_l$; $d_R = d_r$; $e_{min} = e_{lr}$;
    end if
  end for
end for

return d_L, d_R

Table 4.1: The grasping success rate on different types of clothing.

Successful Rate	t-shirts	shirts	sweaters	jeans	jackets	average
categories	90.0%	78.3%	93.3%	76.7%	85.0%	84.7%
items	95%|85%|90%	70%|80%|85%	95%|95%|90%	70%|75%|85%	80%|95%|80%	

4.6.1 Garment Grasping Experiments

In this section, the grasping performance of the proposed approach is evaluated. The evaluation of robotic grasping has two parts: firstly, the success rate of single-trial grasping is investigated; secondly, the effectiveness of grasping is evaluated by counting the required number of trials for completing a successful grasping.

Single-Shot Grasping Experiment

In the first grasping experiment, the grasping performance among five categories including t-shirts, shirts, sweaters, jeans and jackets are tested. Each category has three items of clothing, and 20 grasping experiments are tested on each item of clothing (in total 300 experiments). In each grasping experiment, the selected item of clothing is initialized to an arbitrary configuration by grasping and dropping it on the table. A successful grasping case means that: the gripper is moved to the position indicated by the visual feature; the grasping pose fits the shape of the region to grasp; and the clothing is fetched up. Since this work is

Table 4.2: The required number of grasping trials for a successful grasping on different types of clothing.

Number of Trials	t-shirts	shirts	sweaters	jeans	jackets	average
categories	1.1	1.23	1.1	1.27	1.17	1.17
items	1|1.2|1.1	1.2|1.2|1.3	1.1|1.1|1.1	1.4|1.3|1.1	1.1|1.1| 1.3	—

focused on visual perception of grasping rather than kinematics, in these experiments, the flatness-priority grasping is attempted first, while if the inverse-kinematics cannot be solved, the height-priority strategy is then applied.

The experimental results of the first grasping experiment are shown in Table 4.1. Overall, the grasping success rate varies from 76.7% to 93.3% on different types of clothing. This difference can be attributed into the difference of clothes materials. In other words, the thickness and stiffness variation of clothes materials brings different difficulties to grasping. More specifically, the sweaters and t-shirts are of the best performance (93.3%,90%) while jeans and shirts are of the worst (78.3%,76.7%). The reason is two-fold: firstly, the more stiff the clothing material is, the more difficult the grasping is; and also, the more wrinkles the clothing configuration has, the easier the grasping is. On average, the proposed method is able to achieve 84.7% success rate among the five categories of clothing. In addition, the grasping performance on each item of clothing is also shown in the table. All 15 items of test clothing can achieve at least 70% successful grasping rate.

Multiple-Trial Grasping Experiment

Apart from the single-trial grasping success rate, the other criterion required to be evaluated is the number of trails for completing a successful grasp, which shows how effectively this proposed grasping approach is in handling difficult configurations. In the implementation of this work, the proposed grasping feature provides a ranked array of grasping candidates, then the robot attempts to grasp them sequentiality until the grasp is completed successfully. In order to acquire the grasping status, tactile sensors are used to detect whether the gripper is holding the garment. The experimental results are shown in Table 4.2, in which 150 successful grasps are completed (10 experiments on each item of clothing) and 1.17 trails are required for each successful grasp on average. As shown in the table, similarly to the first grasping experiments, stiff clothes such as jeans and shirts require more grasp trails (1.27 and 1.23 times, respectively). The robot requires the least number of grasp trails on sweaters and t-shirts (1.1 and 1.1 times, respectively). The deviation between different items of clothing is small; the required number of trails ranges from 1.0 to 1.4 among all of the items of clothing. Among these 150 successful grasps, only 1 grasp is completed with 4 trails, 3 grasps with 3 trails, 17 grasps with 2 trails, and the other 129 grasps are completed

with only 1 trail.

Overall, the experimental results of the proposed grasping method demonstrate a reliable grasping performance in terms of its grasping success rate (84.7%) and its effectiveness of grasping difficult configurations[10] (1.17 trails on average).

4.6.2 Garment Flattening Experiments

This section evaluates the performance of the 'parsing' part of the proposed visual perception architecture and the integrated autonomous dual-arm flattening. This evaluation comprises three different experiments. Firstly, a benchmark flattening experiment comprising eight tasks is established to verify the performance and reliability for flattening a single wrinkle using dual-arm planning (Section 4.6.2). While, in Section 4.6.2, the second experiment demonstrates the performance of the proposed approach while flattening a highly wrinkled garment, comparing the CloPeMa robot stereo head system with standard Kinect-like cameras. Finally, Section 4.6.2 investigates the adaptability of the proposed flattening approach on different types of clothing, in which the performance of flattening towels, t-shirts and shorts are evaluated and compared.

The proposed visual perception architecture is able to detect wrinkles that are barely discernible to human eyes unless close inspection on the garment is carried out. As it is not necessary to flatten these wrinkles, a halting criterion is therefore proposed, which scores the amount of 'flatness' based on the amount of the pulling distance computed in Eq. 4.7. In these experiments, if the flattening distances inferred by the detected wrinkles are less than 0.5 cm (barely perceptible), the garment is considered to be flattened[11].

Benchmark Flattening

Table 4.3: The Required Number of Iterations (RNI) in the experiments.

Benchmark Experiments	exp1	exp2	exp3	exp4	exp5	exp6	exp7	exp8	average
RNI	1	1	1	1	1	1	1	1	1
Dual-Arm Success Rate	100%	100%	80%	100%	0%	100%	100%	100%	85%
Grasping Success Rate	100%	100%	100%	100%	100%	100%	100%	100%	100%

The aim of the first experiment is to evaluate the performance of the proposed flattening method under pre-defined single wrinkle configurations as well as the dual-arm planning performance for flattening in different directions. For this purpose, eight benchmark flattening experiments are performed. As shown in Fig. 4.9, in each instance there is one salient

[10]Here, difficult configurations mean those configurations that lack of graspable positions. They often appear in clothes of stiff fabric e.g. shirt and jeans.

[11]This value is obtained by averaging manually flattened garment examples performed by a human user.

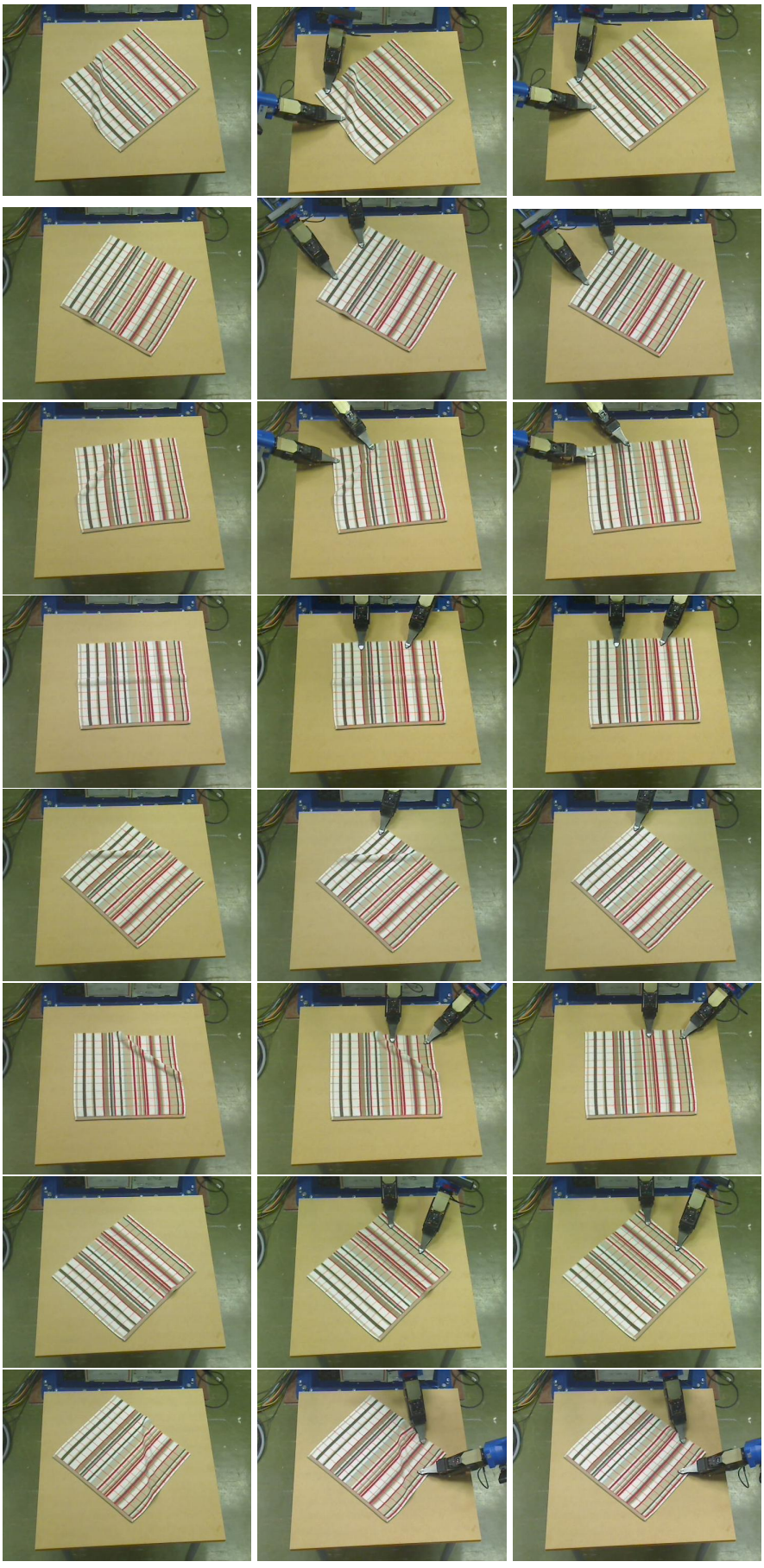


Figure 4.9: Eight benchmark experiments on a single wrinkle using dual-arm planning. Each row depicts an experiment, in which the left images show the stage before flattening; middle, during flattening; and right, after flattening.

Table 4.4: The Required Number of Iterations (RNI) for flattening in highly wrinkled experiments. See text for a detailed description.

Flattening Experiments	exp1	exp2	exp3	exp4	exp5	exp6	exp7	exp8	exp9	exp10	AVE	STD	Dual-Arm Success
Dual-Arm (RH)	4(4)	5(4)	6(4)	5(4)	4(3)	5(3)	4(2)	5(2)	3(2)	6(3)	**4.7(3.1)**	**0.95**	**65.9%**
Dual-Arm (Xtion)	7(4)	8(4)	7(3)	12(4)	8(4)	13(7)	11(3)	10(5)	9(5)	10(5)	9.5(4.4)	2.07	46.3%
Single-Arm (RH)	7	12	5	8	7	7	12	14	8	6	8.6	2.99	-
Single-Arm (Xtion)	10	12	17	11	12	19	13	12	11	14	13.1	2.85	-

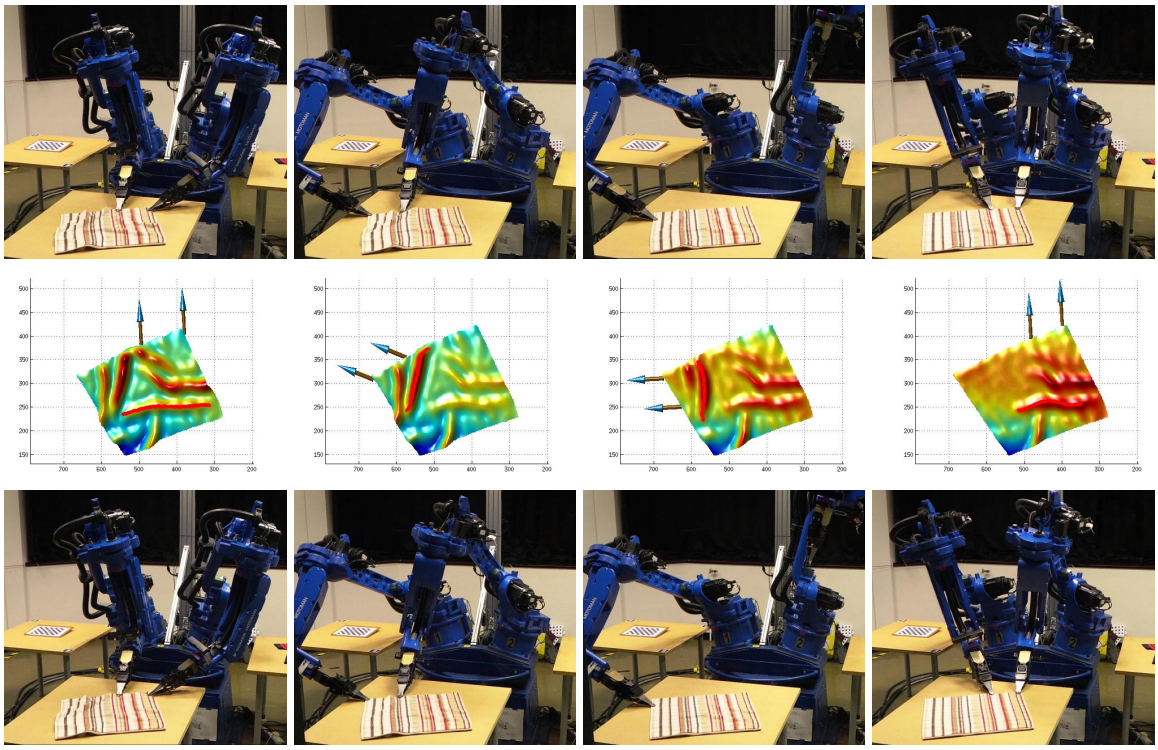


Figure 4.10: A demonstration of flattening an item of highly wrinkled towel. Each column depicts one iteration in the experiment. The top row depicts the towel state before the iteration; middle row, the detected largest wrinkles and the inferred forces; bottom row, the towel state after the iteration. On the third iteration, dual-arm planing demonstrated infeasible to execute, so a single-arm manoeuvre is formulated and applied.

Table 4.5: The Required Numbers of Iterations (RNI) for flattening different types of garments.

RNI of tasks	exp1	exp2	exp3	exp4	exp5	exp6	exp7	exp8	exp9	exp10	AVE	STD
flattening towels	4	5	6	5	4	5	4	5	3	6	4.7	0.95
flattening t-shirt	5	7	12	11	7	8	12	9	12	6	8.9	2.68
flattening pants(shorts)	11	10	5	14	4	3	4	3	2	7	6.3	4.05

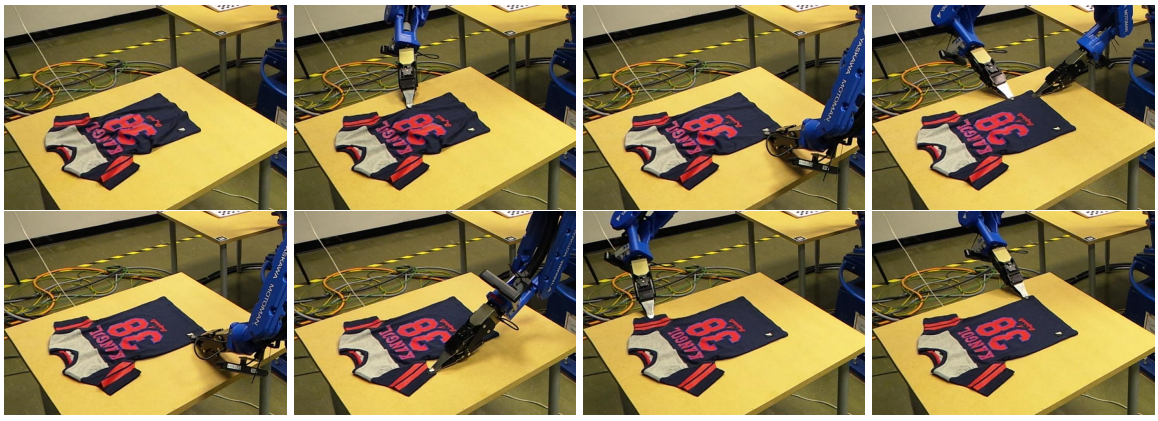


Figure 4.11: An example of flattening a T-shirt. As it is observed, the proposed flattening approach is able to adapt to any shape of garment, the robot can grasp the sleeves and stretch the wrinkles successfully.

wrinkle distributed in the range of 45 degrees to -45 degrees (from the robot's view). In order to obtain a stable evaluation, each experiment is repeated 5 times, and results are shown in Table 4.3.

It can be deduced from Table 4.3 that the proposed flattening approach is able to flatten these eight benchmark experiments with only one iteration. Moreover, the success rate for dual-arm planning is 85%, where the robot successfully grasps the edge(s) of the garment in all of these experiments. Experiment 5 shows a failed case while using both arms; this is because the robot reaches the limitation of its joints and the inverse kinematic planner adopted.

Highly-Wrinkled Towel Flattening

In order to investigate the contribution of the depth sensing provided by CloPeMa stereo head and dual-arm manipulation strategy within the proposed approach in terms of autonomous flattening of highly wrinkled garments, the flattening performance between single-arm and dual-arm strategies is compared. Similarly, in order to demonstrate the utility of high-quality sensing capabilities for the dexterous clothes manipulation, the flattening performance between CloPeMa stereo robot head and a Kinect-like sensor (here the ASUA Xtion PRO is used[12]) is compared.

[12] http://www.asus.com/uk/Multimedia/Xtion_PRO_LIVE/

Therefore, for each experiment, a square towel is randomly wrinkled - wrinkles are distributed in different directions without following an order. Then different flattening strategies are applied (single arm or dual-arm) with either the robot stereo head or Xtion. For comparison, 4 groups of experiments are carried out: (1) dual-arm using robot head, (2) single-arm using robot head, (3) dual-arm using Xtion and (4) single-arm using Xtion. To measure the overall performance and reliability, 10 experiments are conducted for each group of experiment and the required number of iterations (RNI) is counted as shown in Table 4.4. In Table 4.4, each column represents the experiment index for each of the groups proposed above. Values in parentheses show the RNI where dual-arm planning was successful while the rest of the values show the RNI for each experiment.

As shown in Table 4.4, the average RNI for dual-arm flattening using robot head is 4.7 (achieving 65.9% arm planning success rate) while single-arm is 8.6. The average RNI for dual-arm flattening using Xtion is 9.5 (achieving 46.3% dual-arm planning success rate), while single-arm is 13.1. This result shows that a dual-arm strategy achieves a much more efficient performance on flattening than a signal-arm strategy. The standard deviation (STD) of each group of experiments is also calculated, where the STD for dual-arm flattening is 0.95 (using robot head) and 2.07 (using Xtion) while for single-arm is 2.99 (using robot head) and 2.85 (using Xtion). As expected, a dual-arm strategy is not only more efficient but also more stable than a single-arm strategy. Also, from the sensors' perspective, the robot is able to complete a flattening task successfully within 4.7 iterations (dual-arm case) using the stereo robot head as opposed to 9.5 iterations while using Xtion. Overall, the CloPeMa robot head clearly outperforms the Xtion in both dual-arm flattening and single-arm flattening experiments.

The results described above demonstrate that the dual-arm strategy is more efficient in flattening long wrinkles than the single-arm because the latter approach usually breaks long wrinkles into two short wrinkles. Likewise, comparing the CloPeMa stereo head and Xtion, as observed during the experiments, it is difficult to quantify the wrinkles and also estimate the accurate flattening displacement (especially for small wrinkles) from the Xtion depth data because the depth map is noisier than the robot head (the high frequency noise is usually more than 0.5cm). Furthermore, long wrinkles captured by Xtion are often split into two small wrinkles due to the poor quality of the depth map, which in turn results in more flattening iterations (the short detected wrinkles are likely to have a lower dual-arm planing success rate). A video example of the above experiments can be found at: `http://youtu.be/Z85bW6QqdMI`.

Figure 4.12: Ten experiments of flattening t-shorts. Each column demonstrates a flattening experiment, in which the upper image refers to the initial configuration and the lower final configuration.

Figure 4.13: Ten experiments of flattening shorts. Each column demonstrates a flattening experiment, in which the upper image refers to the initial configuration and the lower final configuration.

Flattening Different Types of Garments

Since the proposed flattening approach has no constraints on the shape of the garment, this section evaluates the performance of this method on flattening other types of clothing, namely t-shirts and shorts. Ten flattening experiments are performed for each type of clothing. Examples are shown in Fig. 4.12 and Fig. 4.13, respectively.

The RNIs of different clothes categories are shown in Table 4.5, and here the towel flattening performance is presented as the baseline performance. As shown in the table, towels require an average of only 4.7 iterations to complete flattening. Shorts need more iterations on average(6.3) and t-shirts require still more (8.9). The reason is that towels are of the simplest shape among these tree categories of clothing, while the shape of shorts is more complicated and that of t-shirts is the most complex. This experimental result demonstrates that the proposed approach is able to flatten different categories of clothing and that the RNI of flattening clothing is propagating to the complexities of the clothing's 2D topological shape.

4.7 Conclusion

In this chapter, the 'parsing' part of proposed visual perception architecture is reported, and this architecture is integrated with an active stereo vision system and dual-arm CloPeMa robot to demonstrate dexterous garment grasping and flattening. From the experimental validation, the conclusions are: firstly, the proposed generic architecture is able to parse the various garment configurations by detecting and quantifying structures i.e. grasping triplets

and wrinkles; secondly, the stereo robot head outperforms Kinect-like depth sensors in terms of dexterous visually-guided garment manipulation; finally, the proposed dual-arm flattening strategy greatly improves garment manipulation efficiency as compared to the single-arm strategy. The integrated stereo head, visual perception architecture and visually-guided manipulation systems demonstrate the effectiveness of grasping and flattening different types of garments. The experimental results demonstrate that: the proposed grasping approach achieves a satisfactory performance among all the categories of garments; the proposed flattening approach is not constrained to the 2D topological shapes of the different categories of garments; flattening other types of clothing usually requires more iterations than a simple towel since wrinkles are highly crumpled and disordered.

This chapter verifies two hypothesises of this thesis:

- *In order to manipulate a garment, the 3D garments structures can be identified for the grasping or flattening purpose if the garment's local surface shapes are sufficiently understood. In addition, metric information specify the dimensions of these structures must also be recovered through vision in order to determine size compatibility with the end effector being used to manipulation these structures.*

- *By employing multiple perception-manipulation cycles, both robotic perception and manipulation goals can be incrementally approached in the integrated autonomous robot system (this process can be non-monotonic[13]).*

On the one hand, the local 3D shape and topology analysis underpins the localisation and parametrisation of clothes landmarks, and as a consequence, the poses and motions of robot end effectors (grippers) can be indicated to plan dexterous operations to grasp and flatten the localised and quantified landmarks. Moreover, the 3D configuration parsing approach is based on generic surface analysis and tend to fully understand the landmark structures distributed on the clothing surface, thereby demonstrating the adaptability for multiple visually-guided clothes manipulation tasks. On the other hand, the integrated autonomous flattening employs the perception-manipulation cycles, and consequently the clothing configuration is modified towards the flattening goal.

In next chapter, the 'interpretation' part of the proposed visual perception architecture will be introduced, which is applied to clothes category recognition from free-configurations.

[13]Take the flattening process for an example, due to perception faults and manipulation faults, the configuration of the garment can be modified to a worse state (i.e. more wrinkled in flattening), hence this process is treated as non-monotonic.

Chapter 5

Clothing Recognition - Visual Representation

In the following two chapters, the visual perception problem is to recognise clothes categories from highly wrinkled configurations. This problem is not only the key procedure of the clothes sorting system in pre-washing stage, but also a significant computer vision problem – how to interpret 3D highly-deformable objects which are of the quasi-infinite configuration space. In Chapter 5, the 'interpretation' part of the visual perception architecture is presented, which acquires a robust representation to the variations in the 3D configurations of clothing, and in Chapter 6, a Gaussian Process based interactive perception approach is proposed, which maximises the utility of the proposed visual perception architecture.

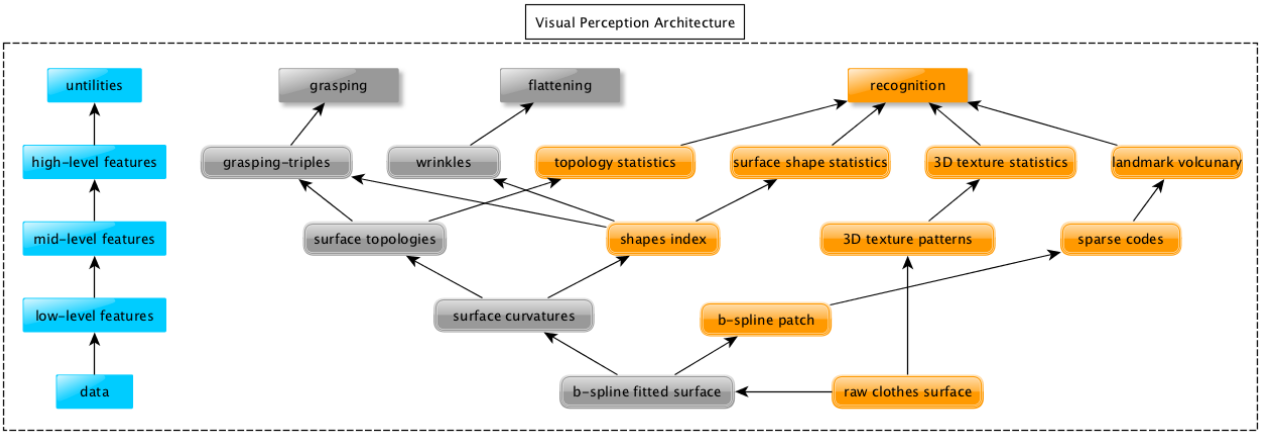


Figure 5.1: The hierarchical visual perception architecture for clothes category recognition.

5.1 Introduction

In real life, the garments are usually in free-configurations e.g. after washing and drying, rather than in a canonical configurations e.g. flat and folded. Recognising clothes categories from any free configuration is a challenging task as the configuration space of clothing is quasi-infinite. Therefore, a clothing configuration interpretation that is robust to the variations in clothing's deformable forms is necessary for this task. From the observations, generic landmarks e.g. wrinkles, folds, 3D textures are widely distributed on the clothes surface in free configurations. Wrinkles and folds are generated by the interactions between clothing surfaces which are determined by the attributes of clothing's materials. While 3D textures are inherent patterns within the textiles. Both of these are robust to the variations in the deformable configurations of clothing.

Inspired by this idea, this chapter presents the 'interpretation' part of the proposed clothes visual perception architecture. This proposed 'interpretation' architecture is able to interpret the 3D clothes surface by describing the generic landmarks i.e. wrinkles, folds and 3D textures; thus it can be adapted to recognise the category of unknown clothing from free-configurations (as shown in Fig. 5.2). More specifically, as it is shown in Fig. 5.1, from the shape analysis, the statistics of surface shape types are obtained; from the topology analysis, the statistics of quantifications of the landmarks i.e. height and width are obtained; from 3D texture analysis, the statistics of patterns of clothes surface are obtained; finally, the 3D shape of each component of landmarks is captured by local 3D descriptors and translated to a vocabulary representation. Technically, local 3D shape descriptors and global statistics descriptors are fused non-linearly, which forms the final robust representation. Moreover, to the best of author's knowledge, this is the first[1] research to integrate clothing category recognition where clothes are in unconstrained and random configurations into an autonomous robot sorting pipeline. To verify the performance of this proposed method, a dataset of 50 clothing items of 5 categories sampled in random configurations (a total of 2100 samples) is created. Experimental results show that the proposed recognition approach achieves 83.2% accuracy while classifying unknown clothing items, which advances the state-of-the-art[Ramisa et al., 2013] by 36.2%. Finally, the proposed approach is evaluated in an autonomous robot sorting system, in which the robot recognises a clothing item from an unconstrained clothes pile, grasps it, and sorts it into a box according to its category. The proposed sorting system achieves a reasonable sorting success rate with single-shot perception.

[1]The existing approaches[Ramisa et al., 2013, Willimon et al., 2013a] are able to recognise clothing categories from unconstrained configurations, while they are invalided in their dataset rather than real-life robot experiments.

5.2 Motivation and Objectives

The motivation of this chapter is to propose the configuration-invariant interpretation of deformable clothing in order to advance the state-of-the-art of clothes recognition. The most commonly reported approaches are devised to recognise clothes categories from canonical configurations (e.g. hanging configurations and flat configurations). However in real laundering scenarios, clothing is usually observed in free form configurations (before or after washing).

The most commonly reported clothes recognition approaches [Kita et al., 2004, 2009a,b, Li et al., 2014a,b, Willimon et al., 2013a, 2011b] are devised for recognising garments from hanging configurations, since the configuration space is greatly reduced. In these recognition scenarios, a large robot with a large working space is required. Hence, medium-sized robots cannot be used to manipulate adult garments. As free-configuration clothing typically presents several occlusions and a much larger configuration space, the performance of existing simulated approaches within these scenarios is limited. Although recognising clothes categories from free-configurations has been explored by [Ramisa et al., 2013, Willimon et al., 2011b], the performance of their reported methods are very limited, and no real-robot experiments are provided. The reason for the limited results of these existing methods is that their proposed representations are not robust to the variations in the deformable configurations and occlusions. More specifically, in [Willimon et al., 2013a, 2011b], the colour information and specific landmarks e.g. pockets, collars, sheaves, etc. are used to interpret the clothes category. Colour information is not robust to the variations existed in different clothing categories because clothing can be rendered in any colour and pattern. In other words, larger-scale training data is required to obtain a stable classifier from those RGB patterns; otherwise it is very likely to lead to over-fitting.

In contrast, depth information is a more robust representation for clothing categories, as clothes are of many colours and textures. Consequently, as compared to RGB-based representation, the required number of training examples is much smaller. Theoretically, one garment can generate quasi-infinite 3D clothing configurations with the same material and type. In practice, the intra-class dissimilarity in terms of depth data still exists, but it is much smaller than with RGB-based data. Moreover, for describing the configuration of clothes, the specific landmarks are sensitive to occlusions and have large variances within categories. In Ramisa et al.'s method [Ramisa et al., 2013], 3D FINDDD descriptors are extracted densely on the clothes surface and the Bag-of-Features method is then employed for coding. The main drawback of this method is that the target of description is not specified; in other words, most of the features are extracted on plain surface rather than on landmark regions. As a result, the codes between different clothing categories are unlikely to be statistically distinctive, therefore very limited performance is achieved.

As mentioned above, the challenges of this recognition task are two-fold: firstly, clothing in free-configurations can be highly wrinkled and occluded, hence it is difficult to encode clothing categories information into generic visual representations; secondly, as clothes are highly deformable objects which have almost infinite possible configurations, learning a generalised category model from limited training data is difficult. To overcome the above challenges, this chapter proposes a generic approach to recognise clothing categories from 3D perception. Instead of detecting the components of clothing such as collars, sleeves, pockets, etc., or describing clothing patterns using RGB-based features, this work focuses on generic landmarks, i.e. wrinkles, folds and 3D textures. For instance, an item of clothing is more likely to be a sweater if it has large wrinkles and folds, and its surface texture is knitted; more likely to be a shirt if it has small and disordered wrinkles and folds, and its texture is jaconet; and more likely to be jeans if it has larger and rigid wrinkles and folds, and its 3D texture is denim. Practically, in order to interpolate the quasi-infinite configuration space, numerous configurations are trained for each item of clothing, and a number of clothing items are trained for each category.

Figure 5.2: Some samples of clothing items from the CloPeMa dataset UG. In this dataset, there are 50 clothing items of 5 categories of different shapes and colours. This dataset is available at: sites.google.com/site/clopemaclothesdataset/.

The contributions of this work are: (1) A generic robot vision architecture for interpreting clothes is presented and integrated within an autonomous sorting pipeline using a dual-arm robot. (2) A novel approach is proposed to recognise clothing categories in random configurations based on high-quality depth data. Experimental results show that the proposed approach advances the state-of-the-art [Ramisa et al., 2013] by 36.2% of improvement in classification accuracy. (3) This approach generalises well and is able to recognise the categories of previously unseen clothing items.

5.3 The Clothes Recognition Pipeline

5.3.1 Outline

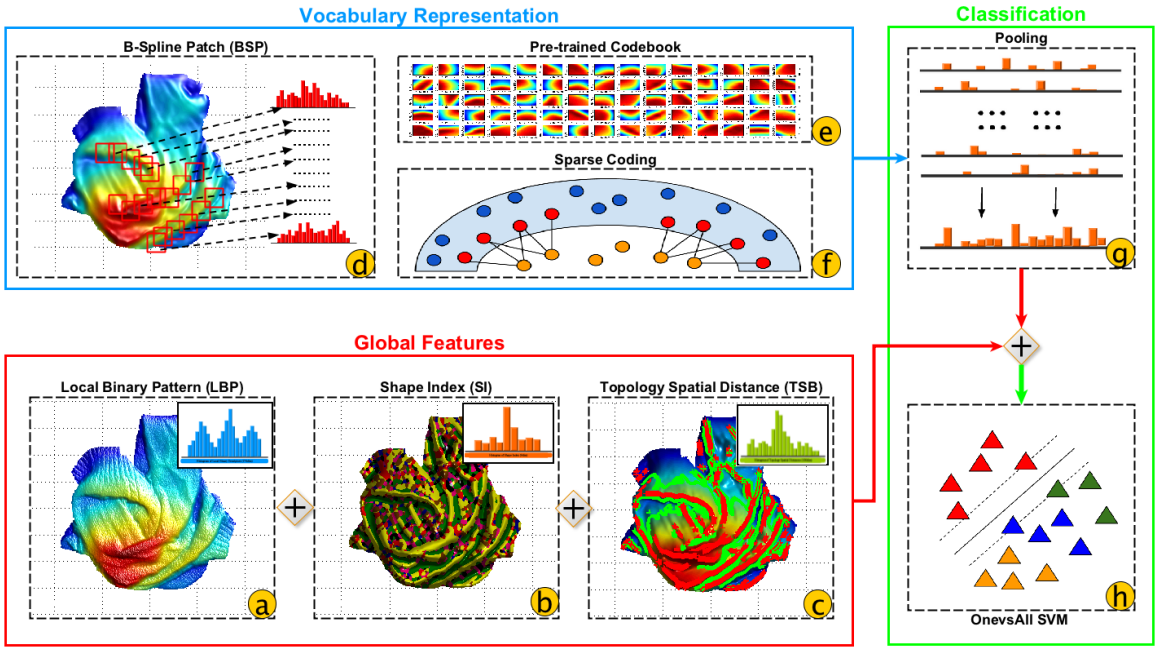


Figure 5.3: The proposed pipeline of clothing category recognition.

The proposed clothes recognition pipeline (Fig. 5.3) consists of four modules: (1) global feature extraction, (2) local feature extraction, (3) encoding, and (4) classification. For global features, *Shape Index* (SI), Local Binary Pattern (LBP) descriptors, and *Topology Spatial Distance* (TSB) are extracted. To be more specific, *Shape Index* features are quantified into a 9 bin histogram to obtain the global statistics of surface shape types of a 3D garment. Moreover, 3D Surface textures of clothing are useful for recognising the fabric types. Among texture recognition approaches, LBP has achieved great success in grey scale texture recognition Ojala et al. [2002a]. In this work, LBP is applied to the high-frequency phase of 2.5D clothes surfaces to describe the 3D fabric structure. Furthermore, this work also proposes a *Topology Spatial Distance* feature computed from surface topologies, which captures the statistics of physical size of garment's landmarks from surface topologies.

Wrinkles and folds are the generic landmarks of deformable clothes especially in free-configuration settings, which are important indicators of clothing categories from the material perspective. Wrinkles and folds can reveal thickness and stiffness attributes of the fabric material thereby inferring the clothes types. For describing these landmarks, local feature, *B-Spline Patch* (BSP) descriptors are extracted densely on surface ridges and then encoded using a sparse coding in order to obtain the vocabulary representation of the 3D

shape of wrinkles. Finally, global and local descriptions are fused together and fed into a discriminative classifier to learn clothing category models.

5.3.2 Generic Clothing Surface Analysis

In Chapter 4, the 'parsing' part of the proposed clothes visual architecture is reported, which details the methodology of parsing a generic 3D clothes surface. In this chapter, the same visual features such as shape index, surface topologies and B-spline surface have been extended from 'parsing' to 'interpretation' (presented in Section 5.3.3 and Section 5.3.4 respectively). For completeness, a brief description of 3D clothes surface analysis is given below.

A piece-wise B-spline surface approximation is adapted to fit a continuous implicit surface onto the original depth map (Fig. 5.3-d). Surface shape and topology features are derived from low-level surface geometries such as curvatures. To detect generic landmarks e.g. wrinkles and folds, and quantify their physical sizes, the surface's topologies are computed, which includes surface ridges and wrinkle contours (shown as red and green lines in Fig. 5.3-c). Ridges are detected by thresholding the maximal curvature at different scales, and the raw ridge lines are filtered by 'ridge' regions (Section 5.3.3). Wrinkle contours are estimated by computing the zero-crossings of the second derivatives of the garment's surface (more details of calculating second order derivatives is shown in Eq. A.12.). Morphological image operation [Lam et al., 1992] is applied to thin ridges and wrinkle contours to obtain ridge lines of one-pixel-width.

5.3.3 Global Features

Histogram of Shape Index (SI)

Shape index [Koenderink and van Doorn, 1992] classifies surface regions into real-valued index values S in the range $[-1, 1]$. The SI value is quantised into 9 intervals corresponding to 9 surface types – cup, trough, rut, saddle rut, saddle, saddle ridge, ridge, dome and cap. Amongst shape types, 'ridge' (shown as yellow in Fig. 5.3-b) is critical in the analysis and description of wrinkles and folds. The shape index value S^p of point p can therefore be calculated as follows [Koenderink and van Doorn, 1992]:

$$S^p = \frac{2}{\pi} \tan^{-1} \left[\frac{k^p_{min} + k^p_{max}}{k^p_{min} - k^p_{max}} \right], \tag{5.1}$$

where k^p_{min} and k^p_{max} are the minimal and maximal curvatures at point p, respectively. In order to parse shape information exhibited by the visible cloth surface, *Shape Index* is computed from the B-spline fitted depth map and apply majority rank filtering (Fig. 5.3-b). This

non-linear filtering removes outlier surface classifications and can be tuned to produce a relatively clean classification of shape types over the cloth surface. For the description of surface shape types, a 9-dimensional histogram of shape index is constructed which interprets the quantifications of 9 surface types. L^2 normalisation is applied on the description.

Histogram of Local Binary Patterns (LBP)

In order to describe the 3D clothing texture, local binary patterns (LBP) descriptors are extracted over multiple scales in the raw depth map of the visible clothing surface. In each scale, the local binary patterns are quantified into a global histogram. That is, LBP histograms are calculated separately at different scales and combined together. In the implementation of this work, a selected collection of patterns (58 patterns) are used [Vedaldi and Fulkerson, 2008]. For a multi-scale feature extraction, a triple-layer Gaussian pyramid is constructed using a sub-division factor of 2 and a Gaussian smoothing parameter ($\sigma = 0.375$). A global multi-scale LBP descriptor of 174 dimensions (58×3) is obtained.

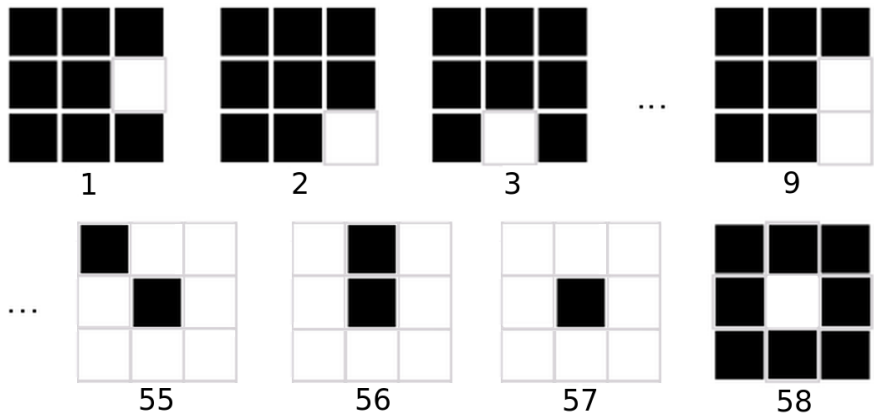


Figure 5.4: The Local Binary Patterns (LBP) used in the proposed method. In theory, there are 256 (2^8) LBP patterns for eight positions. In this thesis, VlFeat's implementation, 58 more representative patterns are selected for more reliable statistics. In this figure, a black block means the depth of its position is smaller than the depth of the center, while a white block means the the depth is larger than the center. The procedure for generating these patterns is: rotating the white block anti-clockwise to get 8 different patterns, then increasing the number of white blocks (1 to 7) and rotating anti-clockwise to get 7×8 patterns in total. Adding two additional patterns, 58 LBP patterns are obtained finally.

Histogram of Topology Spatial Distances (TSD)

Having obtained surface topologies (Section 5.3.2), the physical sizes of generic landmarks can be interpreted by calculating the spatial distance between each ridge point and its nearest contour points. Given all surface ridge points $R = \{r_1, ..., r_{n_r}\}$ and wrinkle's contour points

$W = \{w_1, ..., w_{n_w}\}$ (here R and W are 2D coordinates in the image plane), for each ridge point r_i, the TSD along the $x - y$ plane, TSD_i^{xy}, and depth direction, TSD_i^d, are calculated as follows:

$$TSD_i^{xy} = \min_{i \in n_w} \|r_i, w_j\|^2 \tag{5.2}$$

$$TSD_i^d = d_i^r - d^w_{\arg_j \min \|r_i, w_j\|^2}, \tag{5.3}$$

where d_i^r and d_j^w are the depth value of r_i and w_j. The TSD description is a bi-dimensional histogram of topological spatial distances $\{TSD_1^{xy}, ..., TSD_{n_r}^{xy}\}$ and $\{TSD_1^d, ..., TSD_{n_r}^d\}$, which interprets the statistics of the landmarks' estimated width and height, respectively. The final TSD description is obtained by vectorising the bi-dimensional histogram. Width values smaller than 5 or larger than 50 are removed. For both TSD^{xy} and TSD^d dimensions, bins are set at ten uniform intervals ranging from 1 to 50 (the unit on the x-y plane is pixel, and on the depth axis is millimetre). The latter corresponds to the possible range of wrinkle widths and heights (for TSD^{xy} the unit is pixels, and for TSD^d the unit is millimetres). After applying L^2 normalisation, the final 100-dimensional TSD global descriptor (10×10) is obtained.

5.3.4 Vocabulary Representation

Local B-Spline Patch (BSP)

B-spline surface is a classic 3D surface representation in computer graphics, where the surface shape can be manipulated by a set of control points. In the proposed approach, B-spline surface fitting is adapted to describe local clothing patches. That is, for each $r \times s$ patch P of the depth map, a set of 3D points are given $X\{x_1, ..., x_r\}$,$Y\{y_1, ..., y_s\}$, $P = \{P_{(x_1,y_1)}, ..., P_{(x_i,y_j)}, ..., P_{(x_r,y_s)}\}$, where x_i and y_i are x, y coordinates and $P_{(x_i,y_i)}$ is the value of the depth map. The implicit surface can be represented as [Rogers, 2001]:

$$P(x_i, y_i) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} \Omega_{i,j} \alpha_{i,k}(x_i) \beta_{j,l}(y_i), \tag{5.4}$$

where $\Omega_{i,j}$ represents the control point at row i and column j. $\alpha_{i,k}(x)$ and $\beta_{j,l}(y)$ are the basis functions in the $x - y$ plane (more details of which can be found in Appendix A. Then Eq.5.4 can be written in matrix form as:

$$[P] = [\Phi][\Omega], \tag{5.5}$$

where $\Phi_{i,j} = \alpha_{i,k}\beta_{j,l}$. P is a $r \cdot s \times 3$ matrix containing the 3D coordinates of the range map points, Φ is a $r \cdot s \times n \cdot m$ basis function matrix containing all the products of $\alpha_{i,k}(x)$ and

$\beta_{j,l}(y)$, and Ω is a $n \cdot m \times 3$ matrix of control point coordinates. Thus, the B-spline surface approximation is obtained by solving Eq. 5.5 as a least-squares problem.

$$[\Omega] = [[\Phi]^T[\Phi]]^{-1}[\Phi]^T[P]. \tag{5.6}$$

Finally, the control points Ω are used as the local surface representation, which is a subset representation of the total set of surface points P. In the implementation of this work, a 3rd order uniform open knot vector $[0\ 0\ 0\ 0\ 1\ 2\ 2\ 2\ 2]$ is used to compute the base functions. More details can be found in Appendix A. Each patch is represented by 5×5 control points. Since the control points are distributed uniformly in the x, y plane, only 25 depth values are used for the descriptor. In the proposed method, the BSP descriptors are extracted densely on ridge lines, instead of extracting uniformly across the clothing, since the objective is to describe the 3D shape of generic landmarks (Section 5.3.3).

Locality-Constrained Linear Coding

The Bag-of-Features (BoF) technique [Csurka et al., 2004] only projects a descriptor to its nearest base in the codebook. Sparse coding allows the local descriptor to be represented by more than one codebook base. Compared to traditional L^1-norm sparse coding [Lee et al., 2006], *Locality-constrained Linear Coding* (LLC) [Wang et al., 2010] adopts a locality based constraint to enforce sparsity, which has been shown to perform more effectively and efficiently in many object recognition tasks. In this thesis, the LLC loss function is modified as follows:

$$\begin{aligned} \min_C \sum_{i=1}^{N} \|x_i - c_i B\|^2 + \lambda \|x_i B^T \odot \omega \odot c_i\| \\ s.t.\ c\mathbf{1} = 1, w\mathbf{1} = 1, \forall i \end{aligned} \tag{5.7}$$

where N is the number of local BSP descriptors, $B^{K \times D}$ is the codebook (generated by K-means clustering, D is the dimension of BSP descriptor), and $c_i^{1 \times K}$ is the code for the ith descriptor, $x_i^{1 \times D}$. $\odot$ refers to element-wise multiplication. $x_i B^T$ is the Euclidean distance between x_i and the codebook bases, and $\omega = \{\omega_1, ..., \omega_K\}$ is the weight of the bases, calculated by:

$$\omega_j = \frac{1}{1 + e^{-\sigma(n_j - \bar{n})}}, \tag{5.8}$$

where n_j is the number of descriptors assigned to the jth K-means cluster, and $\bar{n}$ is N/K. In the implementation of this work, σ is set to 0.5×10^{-2} based on practical experience. The motivation of weighting codebook bases is based on the assumption that patterns from

smaller clusters are more distinctive than those from larger clusters. The weights of bases are set as the sigmoid function of the size of corresponding clusters. As a consequence, descriptors will be pushed away from the large clusters and pulled close to the small clusters. The experimental result demonstrates that this codebook base weighting obtains an improvement of 2% - 3% in terms of classification accuracy.

LLC retrieves a very small number $k(\ll K)$[2] of codebook bases that are relevant to the matched descriptor. LLC codes then can be obtained by solving Eq. 5.7. Practically, for each descriptor x_i, once its k nearest dictionary bases are retrieved, the locality constraint $\lambda\|x_i B^T \odot \omega \odot c_i\|$ can be neglected, and the codes C can be approximated by solving the linear problem $\min_C \sum_{i=1}^{N} \|x_i - c_i B\|^2$. Finally, for each obtained sparse code c_i, only k dimensions are non-zero and the sum of $\sum_{j=1}^{K} c_i(j) = 1$. Having obtained the codes of all local BSP descriptors, *pooling* is used to generate the global representation:

$$\begin{aligned} maxpooling:\ c_{out} &= max(c_1, ..., c_n) \\ sumpooling:\ c_{out} &= sum(c_1, ..., c_n) \end{aligned} \tag{5.9}$$

where $\{c_1, ..., n_c\}$ are the input codes and c_{out} is the final output LLC code. The *max-pooling* or *sum-pooling* is calculated for each dimension of the input codes. From practical experience, *sum-pooling* achieves a better performance and therefore is used in the implementation of this work.

5.3.5 Classification

Having all global and local features extracted and fused together, the following step is classification. In this work, the performance of state-of-the-art classification algorithms are investigated, as described in Section 5.4.2. More specifically, SVM with linear and RBF kernels, Random Forest, and Gaussian Process for multi-class classification are investigated. From experimental evaluation, SVM with RBF kernel produces the best performance among the surveyed classifiers. In the implementation of this work, LibSVM [Chang and Lin, 2011] is used and a One-Versus-All strategy is used for multi-class classification.

5.4 Experiments

The objectives of the experiments are two-fold: the first is to evaluate the standalone clothes recognition component in order to evaluate the performance in terms of clothes classification using the proposed interpretation architecture; the second is to evaluate the fully-integrated

[2]From empirical validation, k value is set as 5 according to practical experience.

autonomous sorting solution and examine every procedure of the perception-action cycle. Therefore, the proposed approach is evaluated in two different scenarios: clothes classification experiment and autonomous robotic sorting experiment. The former measures the performance of recognising categories of clothing items from previously unseen clothing items (Section 5.4.2). The latter demonstrates the proposed approach in a dual-arm industrial robot testbed (Section 5.4.3). For the clothes classification experiments, a large-scale RGB-D clothing dataset is captured using CloPeMa stereo head system and the ASUS Xtion Pro (Section 5.4.1). It is worth noting that the proposed 'interpretation' architecture only interprets clothing from depth data, and RGB information is not included.

The clothes classification validation is further divided into 3 experiments: (a) the performance of each local and global features is tested separately, and then the performance of fusing different features is also evaluated; (b) the performance of different classification algorithms classifying the proposed visual feature representation is evaluated; and (c) the recognition performances between using high-resolution stereo data and Asus Xtion data are compared. As reported in [Ramisa et al., 2013, Willimon et al., 2013a], 5-fold cross-validation is adopted as the evaluation mechanism of classification performance, and the classification accuracy is used as the measurement of performance. In this work, each result is obtained by performing the cross-validation procedure ten times and averaging the results.

5.4.1 Clothes Dataset

In order to evaluate the standalone clothes classification performance and perform those proposed experiments, the CloPeMa clothes dataset (University of Glasgow) is captured, which comprises 50 clothing items of 5 categories: t-shirts, shirts, sweaters, jeans and towels. Material types for each category are: fine-cotton, jaconet, wool, denim and coarse-cotton, respectively. Each category has 10 clothing items of various colours (i.e. from white to black colours) and patterns (i.e. from plain to textured). Each clothing item is captured in 21 different random configurations[3] with CloPeMa stereo robot head system and ASUS Xtion.

Consequently, 1050 garment samples are captured in random configurations for each depth sensing device, in a total of 2100 clothing samples. For each item of clothing, an RGB image, depth map and segmented mask are provided, which are of 16 mega-pixels image resolution for CloPeMa stereo robot head and VGA image resolution (640$\times$ 480) for ASUS Xtion Pro. This dataset is the first high-resolution free-configuration clothing dataset available. This dataset is freely available at:
`https://sites.google.com/site/clopemaclothesdataset/`.

[3]There are 20 wrinkled configurations and 1 canonical configuration (flat).

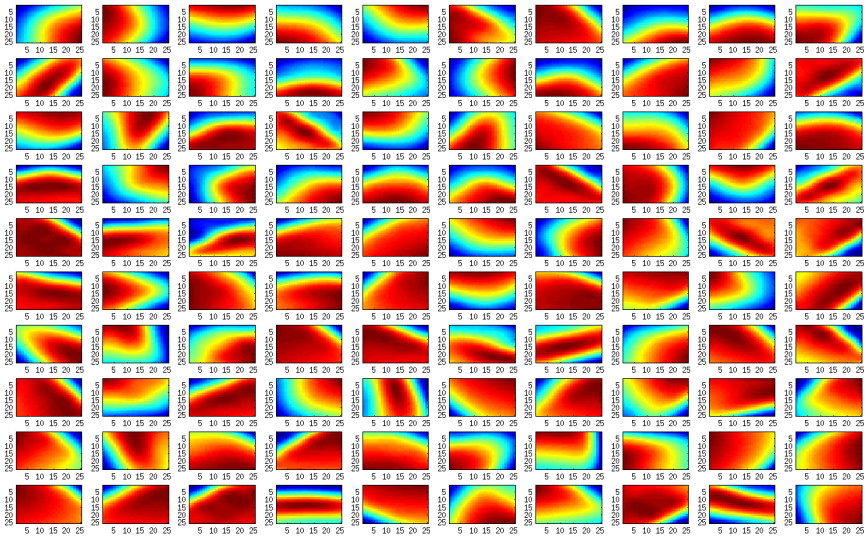


Figure 5.5: Visualised samples of entries of the learnt codebook. They are 3D generic landmarks of the clothes surface.

5.4.2 Clothes Classification Experiments

Baseline Performance

As discussed in Section 5.2, approaches similar to this chapter have been reviewed in the literature. Willimon et al. [Willimon et al., 2013a] used RGB and 3D information of clothes as their visual representation. Hence, their approach is not comparable with this proposed method as the latter's underlying visual representation is based on depth information. While, Ramisa et al. [Ramisa et al., 2013], only 3D information is used for feature extraction and classification. Therefore their implementation is used as the baseline method for comparison. More specifically, for FINDDD descriptor, the number of bins is set to 13, the size of the extraction region is set as 43 for the Asus Xtion and 85 for robot head; and the number of codebook bases is 512. These parameters of FINDDD produce the best performance in the CloPeMa dataset.

The Evaluation of Feature Representation

For local features, K-means clustering is employed to lean a codebook B. For FINDDD and BSP descriptors, 10^5 descriptors are sampled randomly for training. For the BSP descriptor (Eq. 5.5), the learnt codebook can be the visualised by reconstructing B-Spline surfaces (Fig. 5.5). For global features, default parameters (as described in Section 5.3.3) are used since no considerable improvement is recognised when carrying out the experimental validation.

For the first group of experiments, the performance of *B-spine Surface Patch* BSP, *Shape Index* SI, *Topology Spatial Distance* TSD and *Local Binary Patterns* LBP features are eval-

uated independently. For BSP descriptors, LLC coding with sum-pooling is employed. The number of codebook atoms K is set to 256[4]. The confusion matrix using BSP+LLC+sum-pooling is shown in Fig. 5.7(a). In this figure, each column corresponds to a specific category. The values in bottom row indicate the accuracy of its corresponding category. To be more specific, the sum of each row is the number of testing examples. In each column, the diagonal value indicates the number of correctly classified examples and the rest of values are that of incorrectly classified examples.

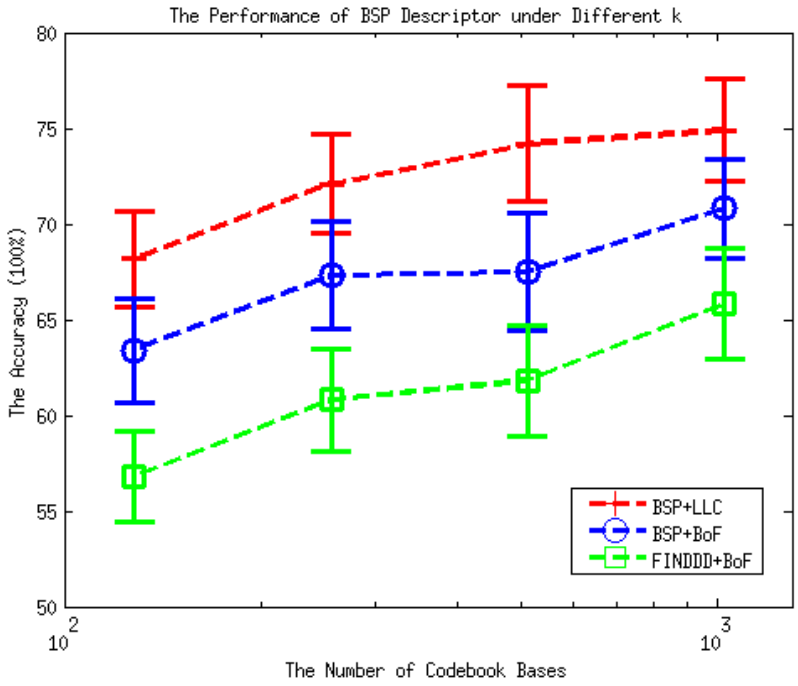


Figure 5.6: The performance of different local descriptors and coding methods. The error bars indicate the standard deviation of (10 times) cross-validation experiments. It is shown that BSP with BoF improves the FINDDD descriptor with BoF by approximately 5%. However, the proposed BSP descriptor with LLC coding improves by 10% compared to FINDDD.

Comparing the results shown in Fig. 5.7(a), 5.7(b), 5.7(c) and 5.7(d), local BSP feature with LLC coding achieves a relatively stable performance among all categories. However, the performances of different global features are inconsistent across specific categories. For instance, LBP performs effectively on jeans and towels but fails on shirts, while SI works on sweaters but fails on t-shirts and jeans. TSD exhibits the best performance on t-shirts (65.7% accuracy) but fails on shirts.

The above suggests that feature fusion is likely to be able to improve the overall performance. To this end, three global features (LBP, SI, TSD) and one local feature (BSP) are

[4]As shown in Figure 5.6, larger K values correspond to better performance. However, a large K value can potentially open up the possibility of over-fitting the training data. Hence 256 codebook bases are used as this value represents a trade-off between over-fitting and generalisation.

TP/FP(%)	t-shirt	shirt	sweater	jeans	towel
t-shirt	**78.7**	14.0	1.1	2.1	4.1
shirt	13.8	**55.9**	7.9	18.4	4.3
sweater	4.3	5.6	**71.1**	7.8	11.2
jeans	1.9	11.8	6.2	**78.9**	1.2
towel	6.5	3.3	11.4	2.7	**76.0**

(a) Local BSP feature + LLC + sum-pooling. The averaging accuracy is 72.1%.

TP/FP(%)	t-shirt	shirt	sweater	jeans	towel
t-shirt	**65.1**	7.4	10.2	0.9	16.4
shirt	15	**56.0**	13.1	15.1	0.7
sweater	11.5	9.8	**61.6**	7.2	10.0
jeans	1.4	6.6	10.8	**80.9**	0.3
towel	7.2	0.1	10.8	0	**81.8**

(b) Global LBP feature. The averaging accuracy is 69.1%.

TP/FP(%)	t-shirt	shirt	sweater	jeans	towel
t-shirt	**57.0**	23	1.6	9.6	8.8
shirt	14.5	**54.5**	6.9	12.6	11.5
sweater	2.3	7.8	**70.0**	17.0	3.0
jeans	3.9	10.4	13.3	**65.9**	6.4
towel	17.6	20.8	8.4	29.5	**23.6**

(c) Global SI feature. The averaging accuracy is 54.2%.

TP/FP(%)	t-shirt	shirt	sweater	jeans	towel
t-shirt	**65.7**	12	2.3	4.5	15.5
shirt	23.1	**33.7**	11.8	16.8	14.5
sweater	2.0	12.2	**45.0**	21.2	19.4
jeans	4.2	19.2	21.7	**47.3**	7.5
towel	9.4	11.1	16.8	3.0	**59.8**

(d) Global TSD feature. The averaging accuracy is 50.3%.

TP/FP(%)	t-shirt	shirt	sweater	jeans	towel
t-shirt	**78.3**	8.2	1.8	0.4	11.2
shirt	11.1	**68.3**	6.0	12.2	2.3
sweater	4.7	5.3	**75.4**	9.2	5.4
jeans	0.5	9.4	8.3	**81.7**	0.1
towel	6.5	1.9	8.0	0	**83.6**

(e) Global features (LBP+SI+TSD). The averaging accuracy is 77.5%.

TP/FP(%)	t-shirt	shirt	sweater	jeans	towel
t-shirt	**89.2**	4.7	1.8	0.2	4.0
shirt	7.4	**70.0**	8.4	13.0	1.1
sweater	2.9	7.0	**80.8**	4.9	4.5
jeans	0.3	8.8	3.8	**87.0**	0
towel	3.4	0.2	7.1	0.1	**88.8**

(f) Fusion of Local and Global features (L-S-T-B). The averaging accuracy is 83.2%.

Figure 5.7: Confusion matrices for multiple class clothes classification. In these confusion matrices, the diagonal elements refer to the percentages of true positives (TF) and the rest elements refer to percentages of false positives (FP).

fused together, yielding L-S-T-B for short. Experimental results are shown in Fig. 5.7(e) and Fig. 5.7(f). It is shown that a fused representation does increase the classification accuracy. Global features (L-S-T) achieve 77.5% accuracy among the 5 categories, where each category is above 64%, while L-S-T-B can reach 83.2% classification accuracy, in which 89.2% is achieved on t-shirts, 70.0% on shirts, 80.8% on sweaters, 87.0% on jeans, and 88.8% on towels. The classification accuracy of shirts is 16% lower than the overall average accuracy. This is because the intra-class dissimilarities of shirts is higher than in the other categories.

The Evaluation of Classification Algorithms

Table 5.1: The comparison between classification algorithms.

Classifiers	Accuracy%
Random Guess	20
FINDDD+BoF+SVM(linear)	56.2
FINDDD+BoF+SVM(rbf)	**61.8**
L-S-T-B+RF	72.0
L-S-T-B+GP(linear)	79.9
L-S-T-B+GP(rbf)	81.0
L-S-T-B+SVM(linear)	80.23
L-S-T-B+SVM(rbf)	**83.2**

This part of the experiments investigates the performance of L-S-T-B representation with different classification algorithms. To that end, this experiment evaluates three state-of-the-art classifiers: support vector machine (SVM), random forest (RF) and Gaussian Process classification (GP). For kernel methods (SVM and GP), both linear and radial basis function (RBF) kernels are investigated. For SVM, the cost parameter C is set as 10, and γ of the RBF kernel is set to $10/D$ (D is the dimension of the L-S-T-B feature description). In order to find the optimal parameters of SVM, systematic cross-validation traverses on large parameter spans ($C \in$[0.1, 100], $\gamma \in$ [0.001 1]). For RF, the forest is initialised with 2×10^3 trees and $D/5$ dimensions are randomly selected for each tree. These parameters are set according to a trade-off between effectiveness and efficiency. In this experiment, the multi-class GP classification is evaluated [Rasmussen, 2006] where Laplace approximation is used as the inference method. Hyper-parameters of GP kernels are optimised by maximising the log marginal likelihood of the training data.

As shown in Table 5.1, the performance of RF (72.0%) is the lowest, using the proposed L-S-T-B representation. SVM provides the best classification performance within the proposed visual representation. For both SVM and GP, the performances of RBF kernels are marginally higher than those with linear kernels. Therefore, SVM with RBF kernel is finally integrated into the recognition pipeline.

Table 5.2: The summary of classification accuracy between sensing devices.

Accuracy	Asus Xtion	Robot head	Improvement
Random Guess	20	20	0
FINDDD+BoF	**47.0%**	**61.8%**	**+14.8%**
BSP+LLC	52.2%	72.1%	+19.9%
LBP	56.4%	69.1%	+12.7%
SI	39.1%	54.2%	+15.1%
TSD	38.6%	50.3%	+11.7%
L-S-B	60.0%	77.5%	+17.5%
L-S-T-B	**64.2%**	**83.2%**	**+19%**

Table 5.3: The performance of autonomous robotic clothes sorting.

Categories	T-shirt	Shirt	Sweater	Jeans	Towel	Overall
Success	7/10	4/10	7/10	7/10	8/10	33/50

The Evaluation of Depth Sensors Performance

In the second experiment, the performance of the proposed clothing category recognition pipeline between sensing devices, i.e. CloPeMa stereo head and Asus Xtion Pro are compared. As shown in Table 5.2, the baseline method (FINDDD+BoF) achieves 47% classification accuracy. The performance of FINDDD is lower than the reported performance in [Ramisa et al., 2013]. This is mainly because the experiments reported in this chapter are about recognising previously unseen clothes rather than recognising unseen configurations of previously seen clothes (reported in [Ramisa et al., 2013]). In contrast, the accuracy with CloPeMa robot head is 61.8% due to the high-quality and high-resolution depth sensing of the CloPeMa stereo head. It can be concluded from experimental results that: firstly, for all features, classification accuracy when extracting from stereo data is approximately 15% higher than extracting from Kinect data; secondly, the proposed visual representation outperforms leading reported approaches on both Kinect and CloPeMa head (64.2% and 83.2% accuracy, respectively).

5.4.3 Autonomous Robotic Sorting Experiments

To demonstrate the robustness of the proposed clothing category recognition pipeline in a real-world scenario, the proposed 'interpretation' visual perception architecture is integrated into an industrial dual-arm robot specifically designed to handle and manipulate clothes. Before the recognition and sorting, a two-stage segmentation approach is applied to segment the whole clothes pile into clothing instances. This is, a grab-cut pre-trained with table's color model[Stria et al., 2014b] is used to segment the clothes pile from the background;

then graph-cut [Felzenszwalb and Huttenlocher, 2004] is employed to segment clothes into instances. In each sorting iteration, the clothes pile is segmented and the largest item (segmentation region) is extracted and feed into recognition pipeline.

In this experiment, 50 items of clothing are put into the validation. They are divided into ten different sorting experiments – clothing items are used only once for each sorting experiment. Similar to cross-validation, those selected clothing items for sorting validation are not used for training. This evaluates the robustness and generalisability of the proposed approach with unseen clothing items. The sorting performance comprises all modules integrated in the robot: image segmentation [Felzenszwalb and Huttenlocher, 2004, Stria et al., 2014b], stereo-matching and stereo-calibration [Cockshott et al., 2012a, Sun et al., 2015] and robotic manipulation [Sun et al., 2015].

As shown in Table 6.6.2, shirts show a low success rate because their intra-class dissimilarity is high. This conforms with the findings in Section 5.4.2. From the author's observation, failures of recognition during autonomous sorting are attributed to: (a) the segmentation algorithm failing occasionally when neighbouring clothing items have similar visual appearance; (b) large clothing items are more likely to be affected by occlusions; and (c) the clothes of soft material properties are likely to be reshaped by other rigid clothes in the pile. Nevertheless, the proposed autonomous sorting requires much fewer perception-manipulation cycles (i.e. one-shot clothing item recognition followed by iterative garment grasping) than reported hang-and-rotate sorting approaches [Li et al., 2014a, Willimon et al., 2011b].

5.5 Conclusion

This chapter presents the 'interpretation' part of the clothes visual architecture and its application for 3D based clothing category recognition from free-configurations. This proposed visual architecture interprets the generic surface attributes from statistical representations of surface shape types, magnitudes of landmarks, 3D surface textures, and vocabulary representation of 3D shapes of local landmarks, which is demonstrated to be a more robust representation than the existing methods.

The recognition performance of the proposed architecture demonstrates a substantial improvement when capturing images with CloPeMa robot head (Table 5.2). This is because CloPeMa robot head can deliver high-quality and high-resolution depth data which underpins a more accurate and detailed 3D surface analysis. Consequently, the performance of the proposed clothes recognition approach advances the state-of-the-art method from 47% classification accuracy [Ramisa et al., 2013] to 83.2% (L-S-T-B). To verify the proposed approach in a real robotic scenario, the proposed recognition approach is integrated into an autonomous clothes sorting robot system. In this system, the robot is able to recognise

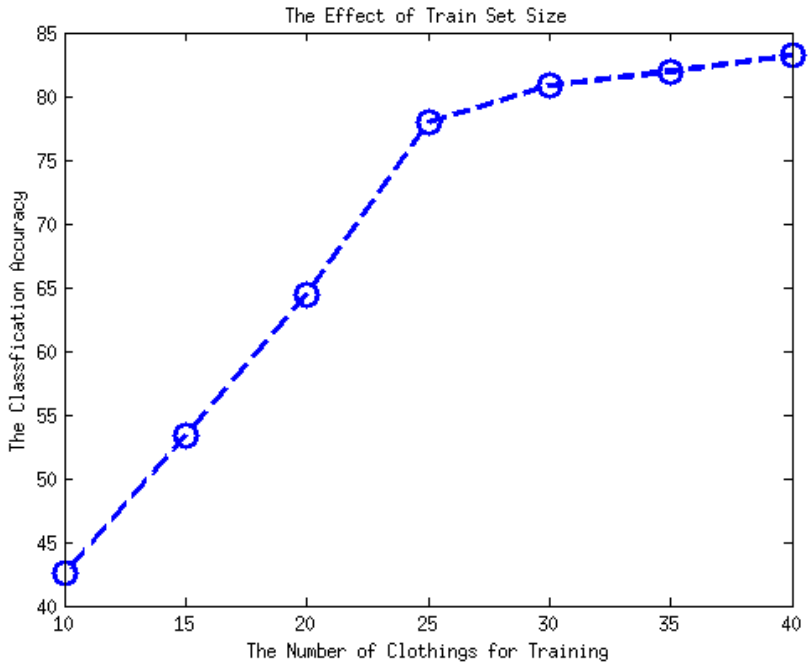


Figure 5.8: In this experiment, the number of training clothing increases from 10 to 40 (for each clothing, 21 configurations are captured). As shown in this figure, the classification accuracy increases along the increase of the number of training clothing.

unknown clothing items in a clothes pile, grasp them and sort them into boxes according to their categories. The whole process is fully autonomous, supported by visual and tactile feedback. To the best of the author's knowledge, this proposed approach is the first integrated autonomous clothes sorting solution based on recognising categories of unknown clothes from free-configurations.

These experimental results verify one of the proposed hypotheses:

- The category of a garment can be recognised from any free-configuration if a robust interpretation that is invariant to the garment's deformable form is proposed. 3D based descriptions of clothing configuration are more robust than RGB based representations for relatively small scale dataset.

The proposed representation describes generic landmarks (i.e. wrinkles, folds, 3D textures) on the 3D clothing surface, which is less likely to be sensitive to the variations in deformable configurations. As a consequence, the quasi-infinite configuration space of clothing can be well interpolated from limited number of training examples, thereby obtaining generalised classifiers for unknown clothes categorisation. Moreover, in this work, using 3D based representation, less than 50 items of clothing are required to obtain a well generalised five category classifier. Fig 5.8 shows the effect of training set size on the classification performance. As a comparison, in RGB based approach [Willimon et al., 2013a, 2011b]'s, over

200 items of clothing are used for training, but a lower performance is achieved. In their clothes recognition approach, 90% accuracy for three categories, 67.54% for four categories and 38.6% for seven categories, are achieved respectively[5]. Therefore, these results demonstrate that 3D based descriptions of clothing configuration are more robust than RGB based representations.

[5]These values are obtained from their paper [Willimon et al., 2013a]. Because different resources of data are used, their method is not evaluated in the experiments of this chapter.

Chapter 6

Clothing Recognition - Interactive Perception

In Chapter 5, the visual perception architecture and a single-shot clothes sorting pipeline are reported. This method is able to recognise the unknown garment from free-configurations with a single-shot perception. However, in the real scenario, there are many ill-posed configurations which are highly-obstructed or very dissimilar from training examples. This chapter focuses on inference and interaction; a novel interactive perception approach is proposed by which the robot is able to 'interacts' with the garment to recognise it. The difference between the single-shot perception and the interactive perception is illustrated in Fig. 6.1. The single-shot perception process is completed after each perception without visual feedback. While in interactive perception, predictive confidence is modelled as the visual feedback, and the robot continues to perceive and interact with the unknown garment until the prediction is confident. This chapter follows the visual representation reported in Chapter 5 but focuses on the classification part of the recognition pipeline, modelling the predictive confidence using probabilistic classifiers.

6.1 Introduction

This chapter proposes a Gaussian Process based interactive perception approach for recognising highly-wrinkled clothes. This recognition method has been integrated into an interactive clothes sorting pipeline for the pre-washing stage of an autonomous laundering process. During the interactive sorting procedure, the robot perceives and manipulates the unknown garment iteratively and calculates the perception confidence until the halting criteria is achieved. This approach differs from previously-reported clothing manipulation approaches by allowing the robot to update its perception confidence via a number of interactions with the garments. The classifiers predominantly reported in clothing perception

studies (e.g. SVM, Random Forest) do not provide true classification probabilities, due to their inherent structure. By contrast, probabilistic classifiers (of which the Gaussian Process is a popular example) are able to provide predictive probabilities. This chapter employs a multi-class Gaussian process classification using the Laplace approximation for posterior inference and optimising hyper-parameters via marginal likelihood maximisation. The experimental results show that the proposed approach is able to recognise unknown garments in difficult configurations and demonstrates a substantial improvement over non-interactive perception approaches.

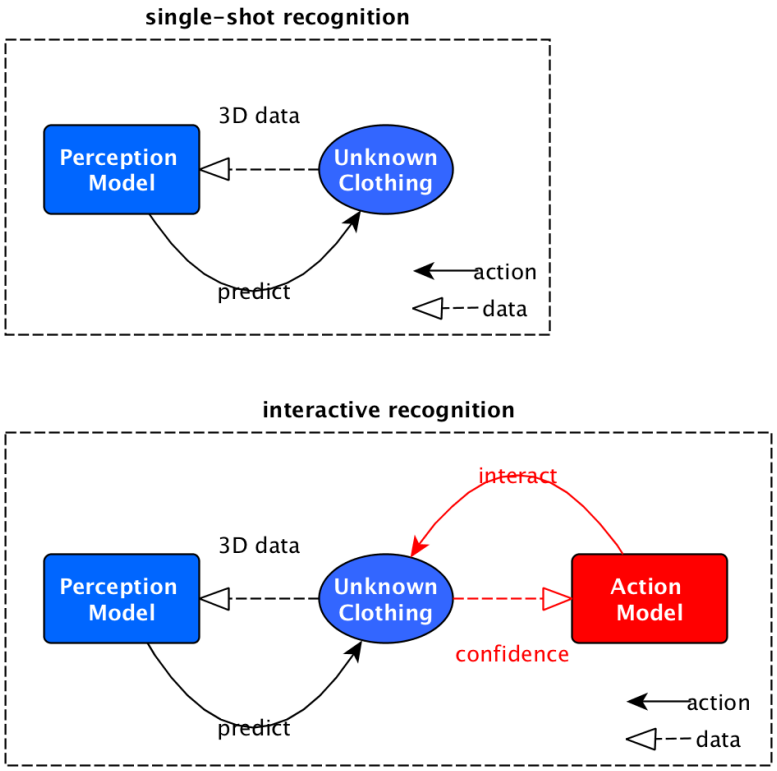


Figure 6.1: The difference between the single-shot recognition and the interactive recognition.

In Section 6.2, the motivation and objectives of this work are given. Section 6.3 presents the visual perception approach of this work and Section 6.4 details the manipulation method for interacting with the garment. The integrated interactive perception pipeline is then introduced in Section 6.5. The experimental results are given in Section 6.6. Section 6.7 concludes this work.

6.2 The Motivation and Objectives

The motivation of this chapter is to overcome the limitations of free-configuration clothes recognition that are observed during the single-shot perception. This chapter proposed to overcome these limitations through the means of interacting with unknown garments. As it is introduced in Chapter 5, the challenges of identifying the clothes categories from free-configurations can be attributed to the huge configuration space presented by garments and the tendency of their surfaces to self-occlude by forming wrinkles or folds. Although the proposed visual perception architecture is able to interpret deformable 3D clothing configuration through an enriched shape and topology based representation, and a certain number of distinctive configurations are trained for each item of clothing, there still exist ill-posed configurations that cannot be interpolated in the infinite configuration space. Therefore, the interaction is required to modify the garment to a recognisable configuration – i.e. interactive perception.

In interactive perception, the most critical problem is to set the halting criteria which determines when to terminate the perception-manipulation loop. For the visually-guided clothes manipulation, the manipulation 'goal' is usually used as the halting criteria. For example, in the clothes manipulation problems reported in Chapter 3 and Chapter 4, the halting criteria of flattening is the threshold of 'flatness' at which the garment is deemed to be flat, while the halting criteria for grasping is the tactile feedback that shows the garment is held by the gripper. However, for the interactive clothes recognition, the goal is to increase the confidence of identifying the clothing type of an unknown garment. For this purpose, this work describes the confidence of the perception that can be used as the halting criterion of the perception-manipulation loop. In general, this confidence can be modelled by the predictive probability or the registration error.

The existed interactive perception approaches [Cusumano-Towner et al., 2011, Doumanoglou et al., 2014b, Li et al., 2015a, Sun et al., 2015, Willimon et al., 2013a, 2011a] have various limitations. For example, non-linear registration is unlikely to be able to match highly wrinkled configurations. The visual or tactile feedbacks used in interactive visually-guided manipulation tasks are unlikely to be adapted to recognition tasks. For classification methods, most predominately-reported approaches have used non-probabilistic classifiers such as Support Vector Machines (SVM) and Random Forests as the classifier [Doumanoglou et al., 2014a,b, Li et al., 2014a,b, 2015a, Ramisa et al., 2012, Willimon et al., 2013a, **?**], or K-Nearest Neighbours (KNN) as the classifier [Willimon et al., 2011b]. SVM classifiers are popular due to their often excellent empirical performance and their use of kernel functions to map data into complex feature spaces in which linear classifiers can be built. However, there are drawbacks to SVMs. In particular SVMs do not provide probabilistic confidences in their classifications. Although the outputs can be post-processed into probability values

[Platt, 1999] this is known to be sub-optimal for small datasets where learning the parameters of the additional probabilistic model is biased by the high proportion of training points that have outputs ± 1 (the support vectors). SVMs are also inherently binary and solving multi-class problems involves combining the outputs of several binary SVMs. Although in some applications that can give high classification accuracy, it is not clear how one should combine the probability values computed for each classifier into a single distribution across the classes. Forest-like classifiers [Doumanoglou et al., 2014b, Willimon et al., 2013a] can generate approximate probabilities via a voting scheme, but the reliability of such estimates is limited by the number of trees and has no formal probabilistic basis.

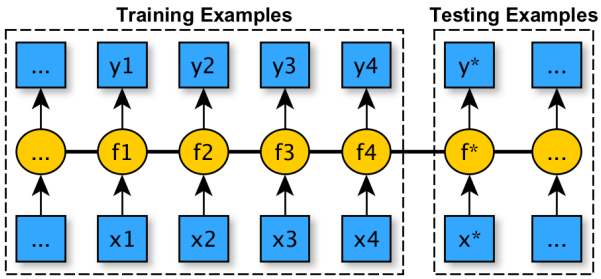
Registration-based methods are capable of matching hanging or sliding-table-edge configurations, and the matching errors can be adapted as the measurement of confidence. However, the performance of registration is more sensitive to the complexity of the garment configurations, which means they are unlikely to be able to match the configurations when subject to high occlusion, e.g. on-table configurations.

In order to mitigate these drawbacks of the commonly-reported classification methods, this chapter employs a multi-class Gaussian Process classification to fully model the distributions within the clothes prediction problem. In this work, the posterior distribution of GP latent variables over training examples are modelled as a multi-variant Gaussian distribution, and then Laplace approximation is employed to estimate it. In addition, the hyper-parameters within the GP kernel are automatically optimised by marginal likelihood maximisation. This work will show that the confidence provided through the conditional probabilities in a probabilistic classifier is a sensible halting criterion for interactive perception.

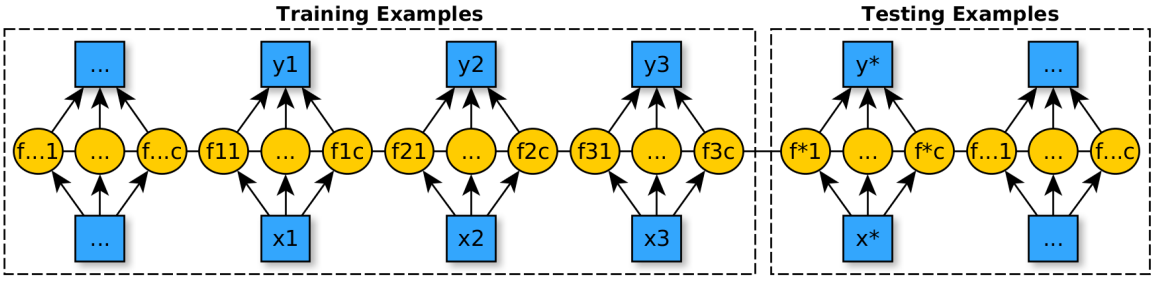
The key contributions of this chapter are two-fold: Firstly, it is the first piece of work to adapt non-parametric multi-class probabilistic classification (via Gaussian Processes) to the clothing recognition problem. Secondly, the proposed GP-based interactive-perception approach is applied to an autonomous sorting pipeline and demonstrated to show substantially improved performance over non-interactive alternatives.

6.3 The Perception Model

This section will introduce the visual perception model for clothing category recognition for highly wrinkled configurations. The proposed visual features are extracted from the depth map produced by CloPeMa stereo head, and it worth emphasising that no RGB information is used in the proposed visual representation of clothing. The visual perception method of this work closely follows the 'interpretation' visual perception architecture reported in Chapter 5, with slight differences on the parameter settings. Further details see Section 6.3.2.



(a) The graph model of basic GP.



(b) The graph model of GP regression and binary classification.

Figure 6.2: The difference between basic binary GP model and the multi-class classification model. In these figures, x refers to examples, y refers to labels and f refers to latent variables. In (b), fij refers to f_i^j, which is the jth latent variable of the ith example.

6.3.1 Stereo Vision System

In this work, CloPeMa robot head with automatic gaze control, camera vergence and GPU accelerated stereo matching is used to deliver a 16 mega-pixel high-quality point cloud stream at 0.2Hz [Cockshott et al., 2012a, Sun et al., 2015]. The advantages of CloPeMa stereo head on dexterous clothes manipulation and clothes recognition are demonstrated in Chapter 4 and Chapter 5, respectively.

6.3.2 Feature Extraction

The high-resolution depth map is able to provide fine details of the garment surface, e.g. tiny wrinkles and 3D textures. However it also increases the computational complexity, suggesting a trade-off between effectiveness and efficiency. In this research, the 16 mega-pixel original depth map is reduced to VGA resolution to obtain quasi-real-time feature extraction. The goal of this work is to advance the performance through interactive perception with fast and inexpensive visual features.

In this approach, the visual perception architecture reported in Chapter 5 is adapted to extract visual representation from depth map in VGA resolution. More specifically, Shape

Index histogram (SI), Topology Spatial Distance (TSD), Multi-Scale Local Binary Patterns (LBP) and landmark vocabulary representation (BSP+LLC coding) are adapted. Shape and topology are the generic attributes of a 2.5D clothing configuration, and LBP describes the fabric patterns. These features are selected as the visual representation of this proposed method because these are robust to various clothes configurations.

Shape index is adapted as one of the global features, in which the shape index values are quantified into 9 bins corresponding to 9 different types of surface. The global topology descriptor (TSD) is also used, in which the distances between each ridge point and its nearest wrinkle's contour point are calculated in the x-y direction and depth direction, respectively. Then the Euclidean distances are quantified into a bi-dimensional histogram. In the implementation of this proposed work, 10 bins ranging from 1 to 20 pixels in x-y direction and 1 to 50 millimetres in depth direction with uniform interval are used, and the dimension of the final TSD descriptor is 100. The details of shape and topology analysis can be found in Chapter 3. To describe the 3D fabric texture, global LBP histograms are extracted in multiple scales from the raw depth surface. In the implementation of this work, vlfeat's [Vedaldi and Fulkerson, 2008] selected 58 patterns are used, which are extracted in three scales of Gaussian pyramids (174 dimension in total). All the global features undergo L^2 normalisation before constituting the final representation.

For highly wrinkled configurations, wrinkles and folds are the landmarks of clothes configuration. In order to describe the shape of wrinkles and folds, B-Spline Patches (BSP) are adapted as the local 3D descriptor and extracted densely from the detected wrinkles and folds. As shown in Fig. A.1-d, B-Spline surface can be obtained through multiplying sparse control points and basis functions. For a B-spline surface approximation problem, given a raw depth patch, the control points can be obtained by solving the inverse problem. The knot-vector used in this work is [0 0 0 0 1 2 2 2 2] and patch size is 21×21. As a result, 5×5 control points can be obtained for each patch, and as these control points are uniformly distributed, only the 25-dimensional depth values are used as the BSP descriptor. In the implementation of this work, locality regularisation is used to normalise each BSP descriptor, and this can be achieved by subtracting its mean value. After extracting the local BSP descriptors, Locality-constrained Linear Coding (LLC) is employed to encode the BSP descriptor with an off-line trained codebook and sum-pooling is used to get the image-level description. In the implementation of this work, in LLC, the five nearest atoms are retrieved as local coordinates. The number of codebook atoms is 256, and K-means is used to train the codebook.

After the global and local features are extracted, the final representation is obtained by fusing LBP, SI, TSD and BSP (L-S-T-B) features. The final description is 539 (9+100+174+256) dimensions.

6.3.3 The Gaussian Process Model

This work combines the advantages of kernel-based approaches (as found in SVMs) with a principled probabilistic framework by using multi-class Gaussian process (GP) classifiers. In contrast to the SVM, the GP is probabilistic by default and can provide us with probability distributions across all clothing categories. In this work, multi-class GP classification is used, with the standard Laplace approximation to estimate the posterior (to overcome the non-conjugacy of the likelihood and GP prior) and covariance hyper-parameters optimised by maximising the log marginal likelihood. This proposed approach closely follows that described in [Rasmussen, 2006] (Chapter 3 and 5) where their hyper-parameter optimisation is extend from the binary case to the multi-class case. Unfortunately, in GPML's toolbox, only binary classification is provided. Although multi-class classification can be solved by one-vs-all or one-vs-one voting using binary classifiers, the class-conditional distributions within multi-classification problem are unlikely to be well-modelled. Therefore, the multi-class GP classification with hyper-parameter optimisation is implemented for this research[1].

Firstly, the rules of the symbol usage in this chapter are illustrated here. Unless otherwise mentioned, upper-case denotes matrices and lower-case denotes vectors. The symbols with subscript and superscript '*' refer to testing examples. Other subscripts and superscripts refer to vector and matrix indices.

Before giving the details of GP multi-class classification, a short introduction about GP is presented. GP is a non-parametric model in which the model is a collection of latent variables (numeric values). For a basic GP model problem, as shown in Fig. 6.2(a), each example has a latent variable. Given the training examples X and testing example x_* and their latent variables f and f_*. For the regression problem, the latent variable refers to the mean of the target, and for the classification (binary case) problem it can be squeezed into sigmoid function to get probabilities of categories. In the regression problem, in order to predict a testing example, the latent value of the testing example can be estimated by the conditional probability of f_* given the joint distribution on training and testing examples $f_*|X, x_*, f$; these can be straight-forwardly calculated since the joint distribution of f and f_* is $\begin{smallmatrix} f \\ f_* \end{smallmatrix} \sim \mathcal{N}(0, \begin{bmatrix} K_{XX} & K_{Xx*} \\ K_{x*X} & K_{x*x*} \end{bmatrix})$. The GP classification problem is introduced below.

In the classification problem, the GP classifier fits a real-valued latent variable to each observation. Jointly, the set of latent variables are given a GP prior (which typically enforces a degree of smoothness for the latent function over the input space). The classification probabilities are obtained by pushing the real values through a squashing function (e.g. the sigmoid function, soft-max function). The training phase consists of obtaining a posterior density over the latent function $p(f|X, y)$ (y is the training labels). Prediction consists of using this posterior to perform a regression to give the latent values at testing points, which are then

[1] https://kevinlisun@bitbucket.org/kevinlisun/multi-class-gpc.git

squashed to provide predictive probabilities. To extend the GP to multi-class classification, one latent function is fitted for each of the C classes. The classification probabilities are obtained by squeezing the C function values for each observation through a soft-max function. To make predictions for a test point, C regressions are performed (one with each of the latent functions) and the resulting probabilities are pushed through the soft-max. The multi-class GP classification is detailed as follows.

In particular, given N training examples (with $\{n_1, n_2, ..., n_c\}$ examples in each class, $\sum_i n_i = N$), $X = \{x_1^1, ..., x_{n_1}^1, x_1^2, ..., x_{n_2}^2, ..., x_1^C, ..., x_{n_c}^C\}$, and corresponding labels are denoted $y = \{y_1^1, ..., y_N^1, y_1^2, ..., y_N^2, ..., y_1^C, ..., y_N^C\}$ where $y_i^c = 1$ if the ith example belongs to the cth class. This vector is therefore of length $Cn = C \times N$. In the description of this work, following [Rasmussen, 2006] the C sets of latent variables (each of length N) are concatenated into one Cn-length vector, f.

Ultimately, the prediction of the class of an unknown instance x_* needs to be solved as ([Rasmussen, 2006]):

$$P(y_*^c = 1|x_*, X, y) = \iint P(y_*^c = 1|f_*)p(f_*|f, x_*, X)p(f|X, y)\, df_*\, df. \tag{6.1}$$

Now each of the terms on the right hand side is analysed in turn. The first term is the standard soft-max function:

$$P(y_*^c = 1|f_*) = \frac{\exp(f_*^c)}{\sum_j \exp(f_*^j)}, \tag{6.2}$$

where f_* is used to denote the C latent variables for the unknown instance. The second term in Eq. 6.1 is a standard noise-free GP regression. Defining the GP prior of this work with a zero mean function and kernel matrix K: $f|X \sim \mathcal{N}(0, K_{XX})$, and defining k_{x_*X} as the $1 \times N$ vector of the kernel function evaluated between the test point and all of the training points, and $k_{x_*x_*}$ as the kernel scalar evaluated at the test point, this is:

$$f_*|x_*, X, f \sim \mathcal{N}(k_{x_*X}K_{XX}^{-1}f,\ K_{x_*x_*} - k_{x_*X}K_{XX}^{-1}k_{Xx_*}). \tag{6.3}$$

In multi-class classification of GP, the covariance matrix K_{XX} is a $Cn \times Cn$ diagonal matrix consisting of C of $n \times n$ covariance matrices $\{k_{XX}^1, \cdots, k_{XX}^C\}$ on the diagonal corresponding to C classes. Similarly, K_{x_*X} and K_{Xx_*} are also diagonal matrices. The final term in Eq. 6.1 is the posterior density over the latent function for the training examples. In classification problems, this is not available in closed form, and the popular Laplace approximation Williams and Barber [1998] is used. This approximates the posterior with a multiple-variate Gaussian (in this case, a Cn dimensional Gaussian) centred at the maximum of the posterior

and with covariance equal to the negative inverse of the Hessian matrix at the maximum.

$$p(f|X,y) \approx q(f|X,y) = \mathcal{N}(\hat{f}, -(\bigtriangledown \bigtriangledown \log p(f|X,y)|_{f=\hat{f}})^{-1}), \tag{6.4}$$

where $\hat{f}$ is the value of f that maximises the posterior and $\bigtriangledown \bigtriangledown \log p(f|X,y)|_{f=\hat{f}}$ is the Hessian of the log posterior distribution evaluated at the maximum. The details of Laplace approximation are shown in Appendix B.

Given the three terms in Eq. 6.1, it is possible to evaluate the integrals to obtain the required predictive probabilities. The conditional probability of f_*, given X, y, x_*, is:

$$p(f_*|X,y,x_*) = \int p(f_*|X,x_*,f)q(f|X,y)df, \tag{6.5}$$

where $q(f|X,y)$ is the Laplace approximation. As both $p(f_*|X,x_*,f)$ and $q(f|X,y)$ are Gaussian distributions (Eq. 6.3 and Eq. 6.4), it is possible to analytically evaluate this integral. The mean of the resulting Gaussian $\mu = \{\mu_1, \cdots, \mu_C\}$ in which each μ_c can be calculated by:

$$\mu_c = (k^c_{x_*X})^T K_c^{-1} \hat{f}^c = (k^c_{x_*X})^T (y^c - \hat{\pi}^c) \tag{6.6}$$

Then, the covariance matrix of the resulting Gaussian is:

$$\Sigma = diag(k_{x_*x_*}) - Q_*^T (K + W^{-1})^{-1} Q_*, \tag{6.7}$$

where W is the matrix containing second order partial derivatives of $\log p(y_i^c|f_i)$ calculated by Eq. B.7. Similar to K_{XX}, Q is the diagonal matrix $diag\{k^1_{x_*X}, ..., k^C_{x_*X}\}$, and $k^c_{x_*X}$ is the vector of covariance between the testing example and training examples with respect to the cth category.

Because of the form of the softmax function, evaluating the integral over f_* is not analytically tractable but is easily approximated via sampling from the predictive distribution over f_*. In particular, if S samples of the C latent variables are drawn, and f_*^{cs} donates sth sample, the predictive probability can be calculated as:

$$P(y_*^c = 1|X,x_*,f) \approx \frac{1}{S} \sum_{s=1...S} \frac{\exp(f_*^{cs})}{\sum_j \exp(f_*^{js})}. \tag{6.8}$$

6.3.4 Hyper-Parameters Optimisation

In the proposed method, the square exponential kernel function (SEiso) is used:

$$k_{SEiso}(x_1, x_2) = \alpha^2 \exp^{-\frac{1}{2}(x_1-x_2)^T diag(\frac{1}{\beta^2},...,\frac{1}{\beta^2})(x_1-x_2)}, \tag{6.9}$$

where α and β are hyper-parameters of the kernel function. Sensible choice of hyper-parameters is crucial to getting good performance. Following [Rasmussen, 2006], the kernel parameters are optimised via maximising the Laplace approximation to the marginal likelihood (this could also be achieved by a cross-validation procedure). In this work, Broyden-Fletcher-Goldfarb-Shanno [Shanno, 1985] algorithm (BFGS) is employed for the optimisation. Since the BFGS is a derivative based optimisation method, the derivatives of the likelihood are necessary to be computed. Details of the computation of the marginal likelihood and the derivatives can be found in Appendix C. It is worth noting that the inference of the log likelihood derivatives shown in [Rasmussen, 2006] is valid only for binary classification; for multi-class classification, the appropriate inference equations are given in Appendix C. Examples of hyper-parameter optimisation can be seen in Figure 6.3.4. Having optimised hyper-parameters, the multi-class GP classifier can be trained (an example of the learnt latent variables are shown in Fig. 6.3.4) and the unknown examples can be predicted (the obtained predictive probabilities are shown in Fig. 6.3.4, and the prediction labels are shown in Fig. 6.3.4). In this work, the predictive probability is used as the confidence of perception, examples are demonstrated in Fig. 6.3.4.

6.4 The Manipulation Model

For recognising the clothing categories from highly-wrinkled configurations, the manipulation objective is to modify the configuration of the garment and reduce the complexity of the configuration. For this purpose, the possible actions are classified into two discrete actions: *Grasp-Shake* and *Grasp-Flip*, which are also likely to be the most important manipulations in human behaviour.

6.4.1 Action 1: Grasp-Shake

Grasp-Shake reduces the complexity of the garment configuration especially for inside folds. It can be observed from practice that, with the effects of the gravity and the air-friction, the garments are likely to spread out during the free-fall motion.

Graspable candidates will then be found on the selected item of clothing. The heuristic clothing grasping approach is applied to detect and rank graspable positions on the detected

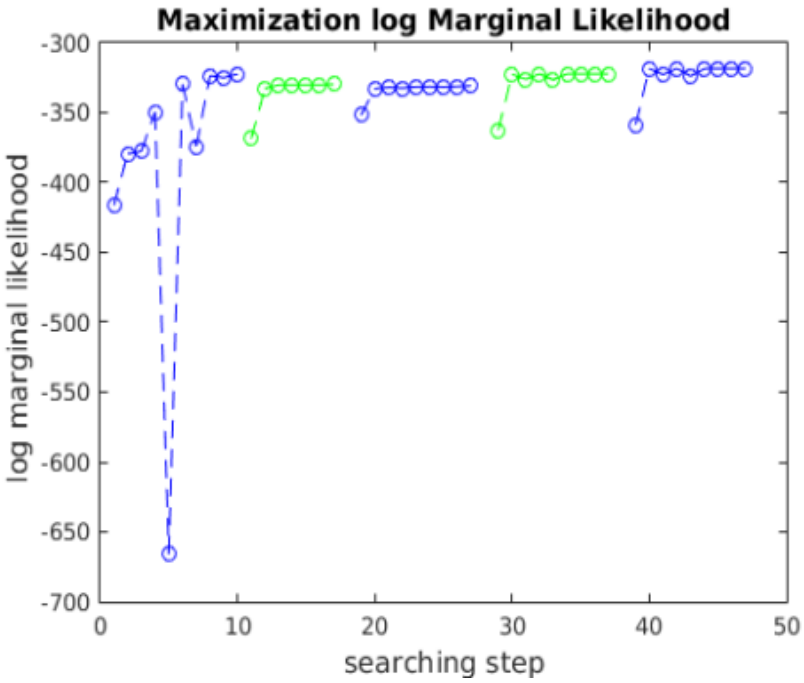


Figure 6.3: The hyper-parameters optimisation through marginal likelihood maximisation. In this figure, the log marginal likelihood is maximized by BFGS. In the proposed approach, multiple initial searching points are adapted in order to avoid suffering from local maximums (shown in different colors).

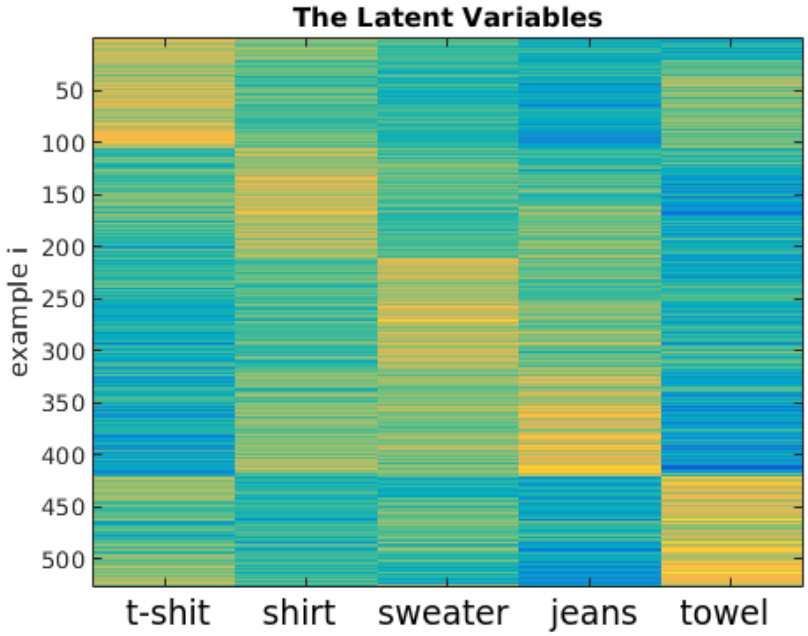


Figure 6.4: The the mean of the latent variables for the training examples (f) estimated by the Laplace Approximation. In the figure, each row refers to an example, and the 5 columns correspond to the 5 categories.

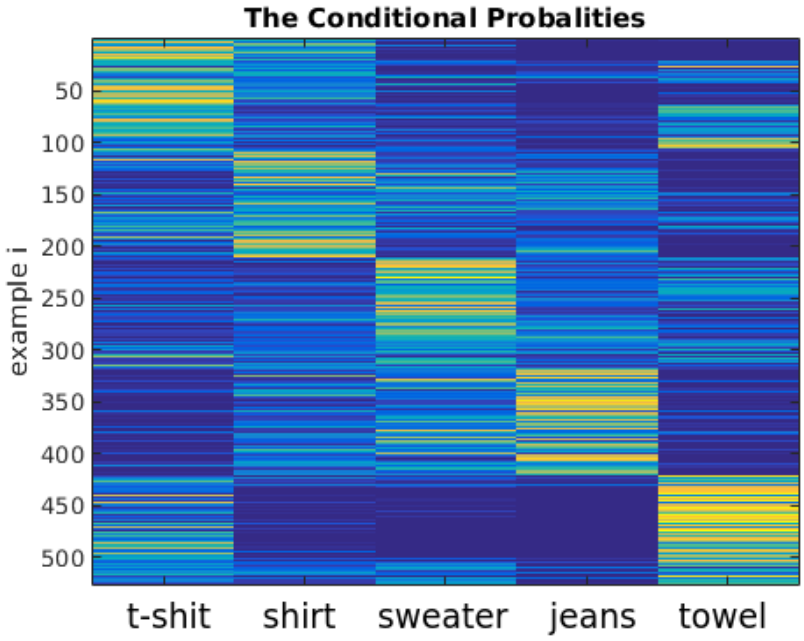


Figure 6.5: The productive probabilities (f^*) for a set of testing examples obtained from multi-class GP classification. In the figure, each row refers to an example, and the 5 columns correspond to the 5 categories.

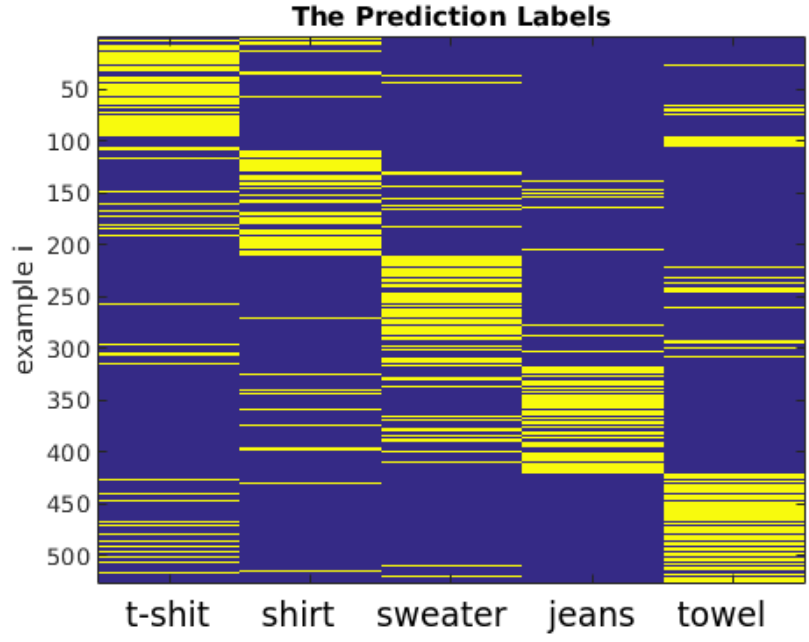


Figure 6.6: The predictive labels for a set of testing examples obtained from multi-class GP classification. This figure presents the final predicted labels, selected by assigning predicting labels to the category for which they have the highest probability. The correct testing labels should be a block diagonal matrix.

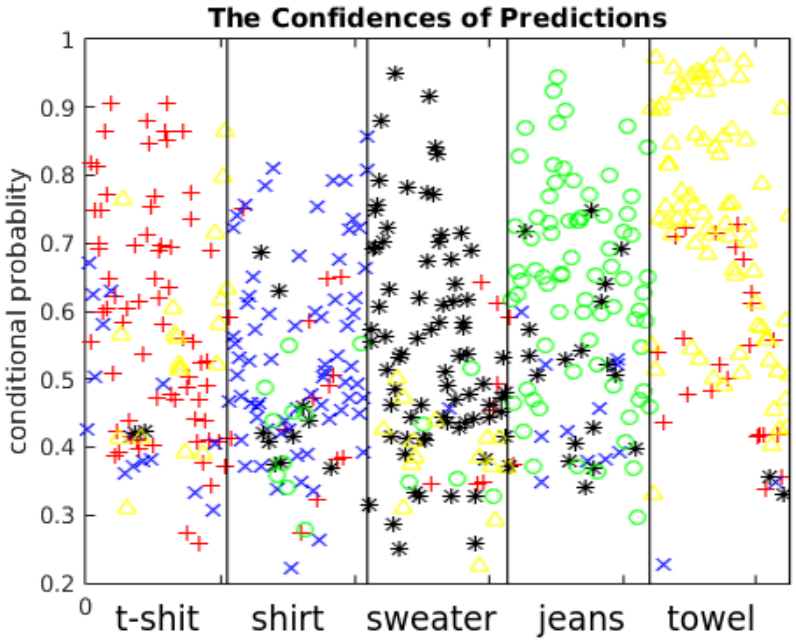


Figure 6.7: The predictions and confidences for a set of testing examples obtained from multi-class GP classification. In this figure, the confidence of the predictions are shown, in which each column corresponds to a clothing category. The correct prediction should be 'red', 'blue', 'black', 'green' and 'yellow', respectively.

wrinkles. More details of detecting wrinkles can be found in Section 4.5.1. During grasping, a success or failure feedback signal is given from the tactile sensor on the tip of the gripper. In the case of failure, other graspable locations are sequentially attempted until the clothing has been grasped successfully.

6.4.2 Action 2: Grasp-Flip

As described in the last section, the occlusion of clothing landmarks is one of the most important difficulties to overcome through interactive perception. In order to observe the hidden side of the garment, *Grasp-Flip* is proposed as the second action, which grasps the garment's edges using a single arm and performs a 'flip' movement to change to field of view of the garment. Similarly to the *'Grasp-Shake'*, with the feedback from the tactile sensor, the robot will attempt to grasp the garment edges in different positions and directions until the grasping is completed. More details of grasping the clothes edges can be found in Section 4.5.2.

6.5 Interactive Perception

From the perception model and manipulation model described in previous sections, the robot is able to, perceive visual features of garments by the proposed 'interpretation' architecture, predict the category labels with predictive probabilities and change garments to different configurations. This work explains how to control the perception-manipulation cycles in the interactive sorting task.

6.5.1 Halting Criteria

The halting criteria, determining when to terminate the interactive perception procedure, is of critical importance in the proposed task. In this approach, the best perception with the most confident prediction is usually adapted as the global confidence, and a threshold δ is used as the halting criteria. Given Pn perceptions:

$$confidence_G = max^C(max^{Pn}(\pi_1, \cdots, \pi_i, \cdots, \pi_{Pn})) \tag{6.10}$$

where π_i are the predictive probabilities of length C obtained by the ith perception. If the $confidence_G$ is larger than δ, the perception is treated as a reliable perception. In the implementation of this work, δ is set as 0.5, which is found based on practical experience as a suitable trade-off between accuracy and time-consumption.

6.5.2 The Interactive Perception and Manipulation Strategy

As shown in Fig. 6.9(a), the workspace of the proposed sorting system comprises: two working tables (the clothes pile is on *table 1* initially; *table 2* is for interactive perception) and five buckets for sorting clothes into. As shown in the autonomous sorting flowchart in Fig. 6.8, the robot starts by capturing and generating RGB-D data. Table 2 is scanned first for garments. If table 2 is empty, the robot turns to find the garments on table 1. If table 2 is not empty, the robot attempts to diagnose the garment. The robot segments[2] the clothes pile on table 1 into instances and attempt to diagnose the garment on the top of the clothes pile. After feature extraction, the features go through GP to generate the predictive probabilities (confidences); and after updating the global confidence, the decision is made whether to sort or keep on perceiving interactively. Meanwhile, the grasping positions are detected for the two proposed manipulations, one of which is chosen depending on the flatness of the

[2]Here the same segmentation approach is used as it is used in Section 5.4.3. That is, a grab-cut pre-trained with table's color model[Stria et al., 2014b] is used to segment the clothes pile from the background; then graph-cut [Felzenszwalb and Huttenlocher, 2004] is employed to segment clothes into instances.

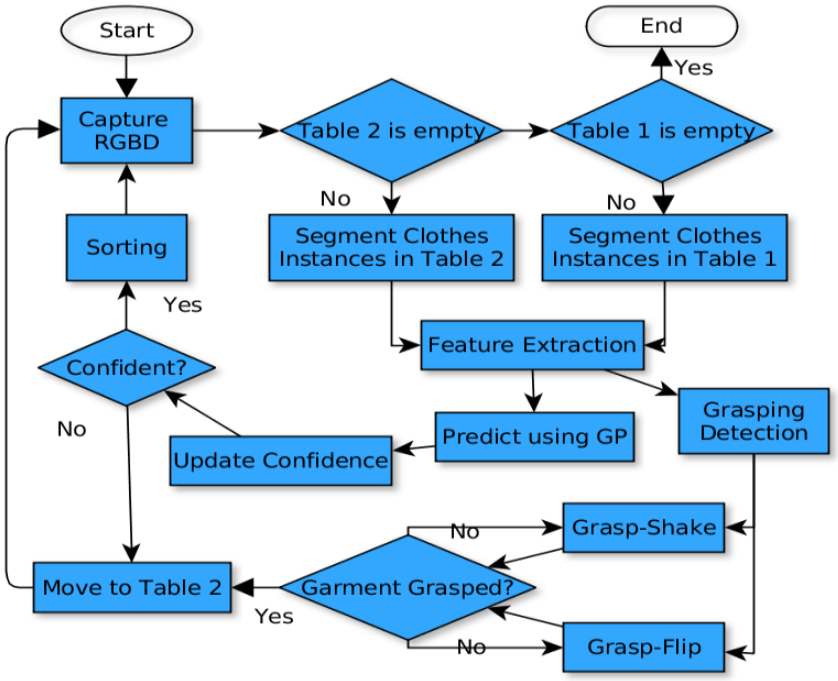
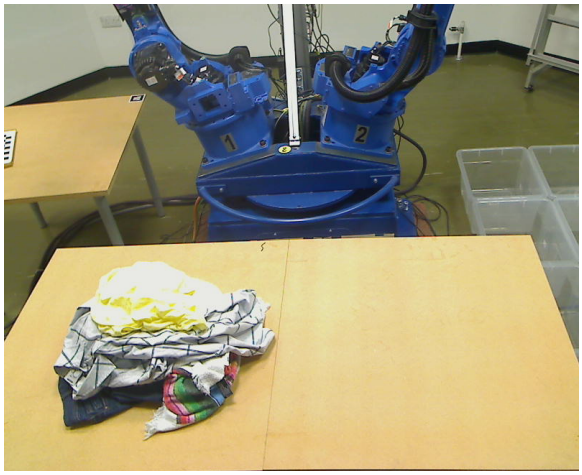


Figure 6.8: Flowchart of the proposed interactive-perception-based sorting system.

garment. Following this strategy, the garment on the clothes pile is interactively perceived on table 2 until the prediction is confident, and the entire sorting task is completed when all the garments of the pile are sorted. A complete demonstration of the proposed interactive sorting is shown in Fig. 6.9.

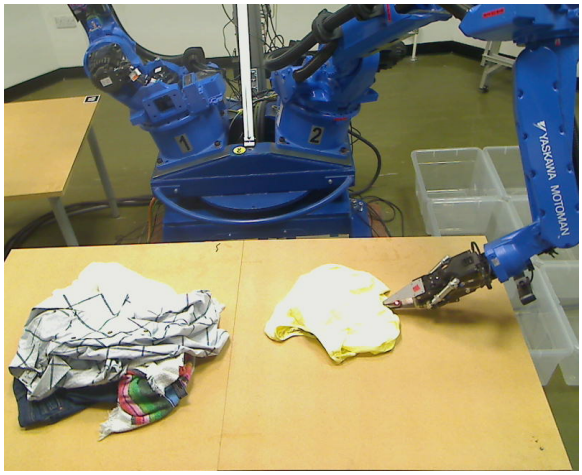
For the two types of manipulation, in the implementation of this work, the *Grasp-Shake* is available if the height of the garment exceeds 5 cm (to avoid collision), and *Grasp-Flip* is available if the thickness of the garment edges is smaller than 5 cm (the maximum opening pose of the robot's gripper). When both of the manipulations are available, the robot makes an arbitrary choice. For *Grasp-Shake*, the 'flatness-priority' grasping strategy is used (detailed in Section 4.5.1), which is more likely to modify the configuration in a larger degree. In *Grasp-Flip*, the robot will attempt to grasp the clothing edge horizontally [3], as this is most likely to flip the clothing upside-down. If the edges on horizontal direction are too high to grasp (more than 5 CM), the robot will keep exploring towards the vertical direction until the available grasping position is found.

[3]The grasping direction is 'from left to right' (in robot head view) from left to right, then changed to 'from right to left' if previous grasping is fault.



(a) Initial stage.

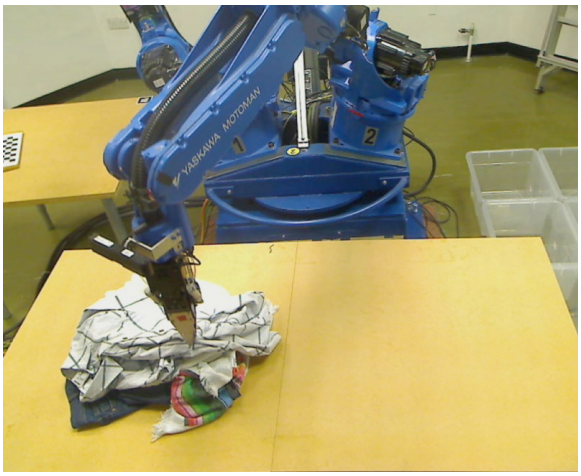
(b) Grasp and shake.



(c) Grasp the clothes edges.

(d) Flip and drop.

(e) Sort into bucket.



(f) Interact with the second garment.

Figure 6.9: A demonstration of the proposed interactive clothes sorting. Table 1 is on the left and Table 2 is on the right. Due to the constraints of the position of CloPeMa stereo head and occlusion of arms, all perceptions need to be performed when the garments are static on the table.

6.6 Experiments

The objectives of the experiments are three-fold: first is to verify whether the predictive confidence is well-modelled; second is to evaluate the performance in a standalone clothes classification, which is taken as the baseline performance; third is to evaluate the performance in autonomous sorting tasks. Following these objectives, the experiments in this work include three parts. Firstly, Section 6.6.1 verifies that, in probabilistic GP classification, the predictions of high confidence are likely to be more reliable. Secondly, the proposed visual perception and GP inference pipeline is evaluated in CloPeMa clothing classification dataset UG (as shown in section 6.6.2). Finally, the performance between the proposed interactive perception method and a non-interactive perception baseline method are compared in an autonomous clothes sorting task (section 6.6.3).

The proposed recognition approach is evaluated in the CloPeMa clothing classification dataset UG[4] reported in Section 5.4.1. Since the focus of this work is inference (classification), 2-fold cross validation is used to evaluate the classification performance. It is worth noting that, in the cross validation, all clothes in the clothes dataset are randomly divided into two sets, one for testing and the other for training. Therefore, the depth map captured from the same item of clothing never appear in both training and testing sets. In other words, the testing examples are absolutely unknown to the classifier.

6.6.1 Validation of Predictive Confidence

The first experiment is to validate the predictive confidence on CloPeMa clothes dataset UG. This experiment investigates whether GP is able to model the conditional probabilities in predicting clothing categories, where the conditional probability of testing examples given training examples, can be treated as the confidence of prediction. Predictions with higher confidences should be more likely to be classified correctly. In order to verify this claim, this experiment analyses the classification performance under different confidence intervals, and the statistical results are shown in Fig. 6.6.1. As shown in the blue curve in Fig. 6.6.1, the classification accuracy experiences a substantial increase when the threshold of confidence internal is increasing. In the red curve, the confidence coordinate is divided into even intervals with the length of 0.1. The accuracy in the confidence interval $[0.2, 0.3]$ is only approximately 0.46; however, it increases dynamically to 1.0 in the interval $[0.9, 1.0]$. The experimental result demonstrates that within the conditional distribution modelled by GP, the predictions with higher confidence are more likely to be correct.

[4]The dataset is available at: https://sites.google.com/site/clopemaclothesdataset/

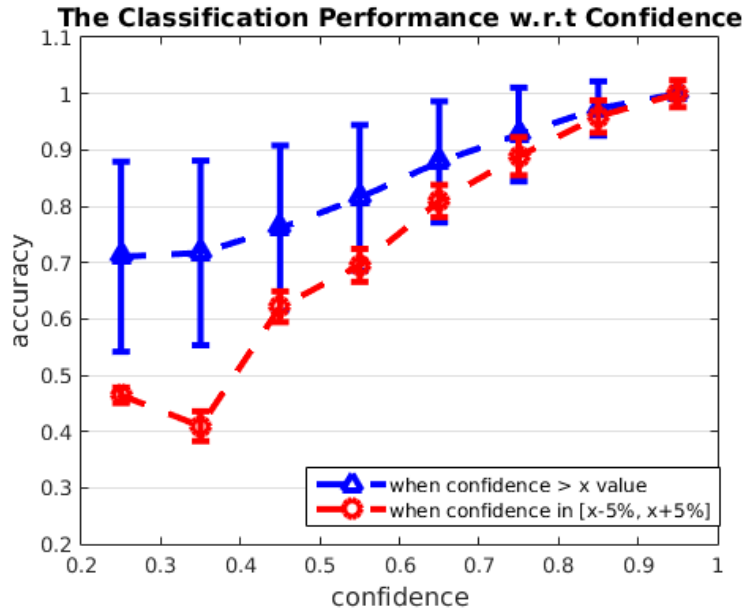


Figure 6.10: The multi-class Gaussian Process classification performance under different confidence intervals. In this figure, the red curve indicates the classification accuracies within the confidence interval $[x-0.05, x+0.05], x \in \{0.25, \cdots, 0.95\}$. The blue curve shows the accuracies where the confidence of prediction is larger than the corresponding x axis value.

6.6.2 Clothes Dataset Experiments

The second part of experiments is to evaluate the classification performance on CloPeMa clothing classification dataset UG. The experiments start with an evaluation of the standalone performance of the proposed visual representation and GP of multi-class classification. The confusion matrix is presented in Table 6.1. In this experiment, Gaussian Process with RBF kernel is used where the hyper-parameters are optimised. As it is shown in the figure, the proposed perception model is able to achieve nearly 72.3% classification accuracy for 5 categories. The accuracies among 5 categories are mostly around or above 70%, except for shirts (52.2%).

Table 6.1: The confusion matrix of clothes classification for 5 categories. The averaging classification accuracy is 72.3%.

TP/FP(%)	t-shirt	shirt	sweater	jeans	towel
t-shirt	**74.3**	11.8	2.3	0.6	11
shirt	15.1	**52.2**	11.9	18.0	2.7
sweater	3.6	6.7	**68.3**	15.0	6.1
jeans	0.5	7.5	7.5	**84.2**	0.3
towel	6.7	1.2	9.0	0.7	**82.5**

Furthermore, the performance with different classification algorithms and visual features are compared. More specifically, two state-of-the-art depth-based visual representations for

Table 6.2: The comparison among different classification algorithms.

Features\ Classifiers	Random Guess	SVM-linear	SVM-rbf	GP-linear	GP-rbf
proposed feature	20	72.4	**73.2**	71.6	**72.3**
FINDDD+BoF	20	44.6	50.7	47.4	48.1
Volumetric Descriptor	20	33.9	36.1	36.8	38.4

Table 6.3: The performance of interactive robotic clothes sorting.

Methods\ Categories	T-shirt	Shirt	Sweater	Jeans	Towel	Overall	Success Rate
Single-Shot Perception	4/10	4/10	4/10	7/10	7/10	26/50	52%
Interactive-Perception	8/10	6/10	7/10	10/10	8/10	39/50	**78%**

clothing recognition (FINDDD [Ramisa et al., 2013] and Volumetric Descriptor [Li et al., 2015b]) are compared with the proposed visual representation. As shown in Table. 6.2, Volumetric Descriptor achieves 38.4% classification accuracy for 5 categories, and the performance of FINDDD is slightly better, approaching 50.7%. The performance of these two description are limited because these are devised for clothes recognition in lightly wrinkled and hanging configurations. The proposed visual representation (L-S-T-B) outperforms the former two descriptions, achieving 73.2% (SVM with RBF kernel) and 72.3% (GP with RBF kernel), taking into account highly-wrinkled configurations, shape, topology and fabric patterns, which are more robust characteristics of garments. Moreover, the comparison between the widely cited SVM and Gaussian Process multi-class classification is investigated. The results are presented in Table 6.2. It can be deduced from this table that the performances of GP are almost as good as SVM; and for both GP and SVM, with a RBF kernel slightly outperforms the linear kernel. In this experiment, the parameters of FINDDD and Volumetric Descriptor are set to default of their implementation because the these parameters are tuned for VGA resolution, the parameters of SVM are optimised based on practical experiences, and the hyper-parameters of GP are optimised by maximising the log marginal likelihood.

The computation complexity and running time of multi-class GP classification are also evaluated: In the training phase, the dominate computation complexity is C^3n^3 times the number of Newton iterations in Laplace Approximation (C is the number of categories, and n is the number of training example); In the testing phase, it is $(C+1)n^3$ for each testing example[5]. In the implementation of this experiment (525 training examples and 525 testing examples), it takes approximately 33.2 seconds for training and 0.11 second for testing (using 12 threads) on an Intel i7 desktop.

[5]It is worth noting that the sampling of posterior distribution (Eq. 6.8) is not included here.

6.6.3 Evaluation of Interactive Perception in Sorting Task

Finally, the proposed interactive-perception approach is evaluated on the robot testbed with autonomous sorting tasks. As a comparison, the proposed visual representation with SVM (RBF kernel) with single-shot perception is used as the baseline method. In this experiment, 50 items of clothing are divided into 10 different sorting experiments; all clothing items are only used once for each sorting experiment. Similarly, for each experiment, those clothing items selected for sorting are not used for training. As shown in Table 6.6.2, the proposed interactive perception approach improves the sorting success rate of the baseline method by 26%. More specifically, the SVM-based single-shot perception only achieves 52% sorting success rate, which is lower than the classification performance in the dataset experiment (73.2%). The reason can be attributed to segmentation faults (clothing instances are not separated), grasping faults (more than one item is grasped) and occlusions. In contrast, the proposed GP-based interactive-perception approach outperforms the dataset classification (72.3%), achieving 78% success rate. From observation, the proposed interactive-perception approach is likely to be able to mitigate segmentation faults and grasping faults because these faults are unlikely to to recur in iterations. For example, in some cases, if two clothing items are segmented as one clothing item, the perception confidence is likely to be low. In the next interactive-perception iteration, the robot will grasp one of these, change its configuration and perceive again. The same segmentation faults are very unlikely to happen again as both the spatial position and configuration of the clothing are modified. More importantly, through interactive perception, the robot is able to reconfigure the clothing to recognisable configurations during manipulations and thereby increase the predictive confidence during perceptions.

6.7 Conclusion

This chapter proposes a Gaussian Process based interactive perception approach to recognise clothing categories from highly wrinkled and highly occluded configurations. Through multi-class GP classification with an optimised kernel adapted to model the distribution of predictive probabilities, the perception confidence for each observation of the unknown clothing under classification can be measured. Therefore, the predictive probabilities of GP classification serve to inform an interaction heuristic as to when sufficient observations of the clothing in new configurations have been accumulated. Accordingly, the perceived confidence of the system, where it exceeds a pre-specified threshold, provides a halting criterion for the proposed interactive-perception pipeline.

The experimental evaluation of the proposed method incorporated within an robot autonomous sorting task demonstrates that interactive perception can not only mitigate the classification

mistakes of ill-posed configurations, segmentation faults and grasping faults prevalent in single-shot perception/manipulation, but also improve perception performance by reconfiguring the clothing under manipulation into recognisable configurations, thereby facilitating the sorting decision.

This chapter validates one of the proposed hypothesises:

- By employing multiple perception-manipulation cycles, both robotic perception and manipulation goals can be incrementally approached in the integrated autonomous robot system (this process can be non-monotonic).

The difficulty of adapting the perception-manipulation routine to clothes recognition is modelling the visual feedback (i.e. perception confidence). Through multi-class Gaussian Process classification, the predictive probabilities can be obtained, to measure the confidence of visual observation, thereby closing the loop in the perception-manipulation cycles. The experimental results show that, by incorporating interactive perception, the utility of visual perception is maximised and the perception goal can be achieved incrementally.

Chapter 7

Conclusion and Future Work

In this chapter, the achievements are summarised, and the limitations are illustrated and the initial hypothesises are reviewed. The scientific achievements underpin the validation of the proposed hypothesises. At the same time that the limitations of the proposed approaches are investigated, the potential solutions to address these limitations are discussed.

7.1 Objectives and Hypotheses Revisited

The objectives of this thesis are: (1) to advance the state-of-the-art visually-guided clothes manipulation through a sufficient understanding of generic 3D clothes configurations[1]; (2) to interpret clothing configurations through an enriched representation which is robust to clothing's deformable form; (3) to integrate advanced robot and object interaction routines for integrated autonomous systems. Corresponding to the objectives, the hypothesises of this work are threefold:

- In order to manipulate a garment, the 3D garments structures can be identified for the grasping or flattening purpose if the garment's local surface shapes are sufficiently understood. In addition, metric information specify the dimensions of these structures must also be recovered through vision in order to determine size compatibility with the end effector being used to manipulation these structures.
- The category of a garment can be recognised from any free-configuration, if a robust interpretation that is invariant to the garment's deformable form is proposed. 3D based descriptions of clothing configuration are more robust than RGB based representations for relative small scale dataset.

[1]here, the generic clothing configuration includes the generic clothing landmarks (i.e. grasping triplet, wrinkles), as well as the shapes and topologies of the surface.

- By employing multiple perception-manipulation cycles, both robotic perception and manipulation goals can be incrementally approached in the integrated autonomous robot system (this process can be non-monotonic).

From the work reported in previous chapters, the objectives have been achieved and the hypothesises have been validated. More details are illustrated in the following subsections.

7.2 Summary of Contributions

This section summarises the achievements of this thesis in four phases: *visual perception architecture*, *visually-guided manipulation*, *clothes recognition*, and *interactive perception*.

7.2.1 Visual Perception Architecture

The major contribution of this thesis is to propose a generic visual perception architecture for robotic clothes perception and manipulation. This architecture is integrated into dual-arm CloPeMa robot to conduct multiple laundering tasks. In all of the existed literature about robot clothes perception and manipulation, the proposed visual architecture is the only one which has been demonstrated and evaluated in a real robot system. This visual perception architecture contains two parts: the *parsing* part for visually-guided clothes manipulation, and the *interpretation* part for clothes category recognition. On the one hand, the proposed visual architecture can contribute to dexterous visually-guided clothes manipulations through parsing the 3D configuration hierarchically, localising and parametrising landmarks precisely. On the other hand, the enriched clothing representation obtained by fusing the statistics of 3D shape, topologies, textures and landmark vocabularies, is robust to deformable configurations. This robust visual perception representation is adapted to clothes recognition in free-configurations and achieves a substantial advancement on the state-of-the-art.

7.2.2 Visually-Guided Manipulation

This thesis presents a visually guided, dual-arm, industrial robot system that is capable of autonomously grasping/flattening garments by means of a novel visual perception architecture, which fully parses high-quality RGB-D images of the clothing scene based on an active stereo robot head. In this proposed pipeline, the state-of-the-art of clothes visually-guided manipulation is advanced in three aspects: depth sensing, garment configuration understanding and dual-arm manipulation skills. To be specific, firstly, instead of using common Kinect-like cameras, this thesis employs active binocular head with GPU accelerated

C3D matcher in order to produce RGB-D data of high-accuracy and high-resolution. Secondly, compared to the most commonly-reported representation focusing on a specific task or with prior knowledge, a more generic 3D configuration parsing based approach is proposed, which localises and parametrises the garment's landmarks precisely. Thirdly, the proposed garment sensing, together with a generic dual-arm flattening strategy without constraints on the garment type, are incorporated into perception-manipulation cycles, which shows the effectiveness and robustness in autonomous flattening tasks.

The evaluations of the visually-guided manipulation include two parts: In Chapter 3, the performance of the proposed garment perception approach is compared with a baseline perception method in physics based cloth simulations by duplicating the same cloth configurations and applying the same flattening heuristic. The proposed visual perception outperforms the baseline approach in terms of flattening performance from 'quality', 'efficiency' and 'stability' perspectives. This is because the proposed clothes configuration understanding approach provides a greater degree of accuracy on wrinkles' localisation and quantification. In Chapter 4, the visual perception architecture with extension to grasping is integrated and evaluated in CloPeMa robot in order to investigate the contribution of depth sensing and manipulation skills. The experimental results show that flattening integrated with stereo head improves the flattening performance (for an average of 4.7 RNI, compared to 9.5 RNI using Kinect-like camera). The proposed dual-arm flattening skill demonstrates success in flattening multiple categories of garments (towel, t-shirt, shorts, etc.). The accurate depth sensing underpins the proposed configuration parsing, and as a result facilitates the dexterous manipulations. While the advanced manipulation skills (dual-arm) can more effectively utilise the observation knowledge for reconfiguring the garment towards the manipulation goals.

7.2.3 Clothes Recognition

In Chapter 5, the 'interpretation' part of the visual perception architecture is proposed and applied to a novel clothes recognition and sorting pipeline. The task is to recognise clothes categories from free-configurations (highly-wrinkled configurations); interpretation of the clothing configuration is based on 3D generic surface analysis. This representation fuses the vocabulary representation of local 3D landmarks and global statistics of shapes, topologies and textures of clothes surfaces. This representation is robust to the variations in configurations of 3D clothes. Within this robust 3D-based representation, the quasi-infinite configuration space of deformable clothes can be interpolated properly from a limited number of training examples.

The proposed visual representation, together with depth sensing, detection, classification and interaction, has been integrated into an autonomous clothes sorting pipeline for the pre-washing stage of the robotic laundering system. More specifically, the RGB-D data is pro-

duced by stereo head, and a couple-layer segmentation is employed to detect the clothes pile on the table and segment it into instances. The enriched representation is then extracted from the largest segmented clothing instance and fed into pre-trained classifiers to get the garment category prediction. Once the garment category is predicted, the garment is then grasped and sorted into the corresponding bucket.

Experimental validations include dataset classification validation and real-robot sorting validation. In the first validation, a large-scale RGB-D clothes dataset is captured (5 categories, 50 items of clothing, 2000 instances) using both stereo-head and Kinect. 5-fold cross-validation is used to evaluate the classification performance on unknown clothes. The proposed approach achieves 83.2% accuracy on recognising unknown clothes, outperforming the state-of-the-art by 36%. This substantial improvement can be attributed to the more advanced depth-sensing and the more robust configuration interpretation. In the second validation, the integrated autonomous sorting pipeline achieves a reasonable performance with real-life sorting tasks (66% sorting accuracy including segmentation, observation, manipulation errors). In all of the pre-existing literature, this is the first autonomous clothes category sorting solution for the pre-washing stage of robotic laundering.

7.2.4 Interactive Perception

In this thesis, most of the robotic autonomous system follows an interactive perception mechanism. As it is shown in Chapter 4, the proposed autonomous flattening approach follows a perception-manipulation routine, tracking the state of the garment, acquiring visual feedback and conducting action. During each perception-manipulation cycle, the latest configuration is parsed by integrated visual architecture, the state of the garment is updated and checked against the pre-defined criteria, and the new flattening manipulation is inferred by generic flattening heuristic. Following this mechanism, the robot is able to flatten the garment incrementally. More importantly, compared to structured manipulation strategies, this interactive perception-based strategy shows better tolerance to incorrect perceptions or manipulations.

A novel interactive perception-based clothes identification and sorting approach is presented in Chapter 6. In all of the existing literature, this is the first probabilistic classification-based interactive perception approach, which remarkably improves the recognition performance through modelling the perception confidence. In this work, the perception confidence is represented via the 'true' predictive probability generated from multi-class Gaussian Process classification. In the integrated interactive sorting system, the perception model acquires data, extracts visual representations and makes predictions for the observation. Then the confidence of prediction is fed into the manipulation model to decide whether to interact or sort the garment. This proposed interactive sorting approach is able to overcome ill-posed configurations and to eliminate other errors (e.g. segmentation, manipulation errors) existing

in the system. The experimental results demonstrate that the proposed interactive perception approach is able to boost the accuracy of recognising unknown clothes from 52% to 78%.

7.2.5 Summary

This thesis proposes a generic visual perception architecture for a number of clothes perception and manipulation tasks, integrated with several novel autonomous solutions. The proposed visual perception architecture is comprised of two parts: The 'parsing' part is devised to parse and parametrise the 3D configurations for visually-guided manipulation tasks; the 'interpretation' part is to acquire enriched visual representations that are robust to deformable form of clothes. Employing perception-manipulation cycles, the utilities of visual perception can be maximised in the integrated autonomous robot system.

7.3 The Validation of Hypothesises

From the achievements presented in this thesis, the pre-proposed hypothesises can be validated which is discussed in this section.

The first hypothesis of this thesis is:

- In order to manipulate a garment, it is necessary to understand the garment's local surface shapes in order to identify 3D garments structures for the grasping or flattening purpose. In addition, metric information specify the physical dimensions of these structures must also be recovered through vision in order to determine size compatibility with the end effector being used to manipulation these structures.

In this thesis, sufficient 3D shape and topology understanding underpins semantic landmark detecting and parametrising. As it is reported in Chapter 3, an accurate geometry-based wrinkle analysis method (part of the proposed visual perception architecture) outperforms a coarse clustering-based method in simulated flattening tasks, which demonstrates that a more precise localisation and parametrisation of clothes landmarks is able to accelerate the dexterous visually-guided manipulation (flattening). In Chapter 4, garment grasping is achieved by extending geometry-based features to the more generic visual perception architecture. This demonstrates that a full understanding of the garment's local surface structure (wrinkles) is a generic solution for visually-guided clothes manipulations. Moreover, the flattening performances of using Kinect-like camera (AUSA Xtion) and stereo head camera are evaluated and compared, and the experimental result proves that more precise metric information of clothing's structure can be acquired by a better 3D sensing, thereby boosting the performance of dexterous manipulation. In addition, the utilities of manipulation skills are also investigated,

and dual-arm flattening outperforms single-arm flattening due to its advantage on flattening large wrinkles.

The second hypothesis of this thesis is:

- In order to recognise a garment from any free-configuration, a robust interpretation that is invariant to the garment's deformable form is necessary. 3D based descriptions of clothing configuration are more robust than RGB based representations.

Instead of describing the garment partially for specific tasks, this thesis proposed a more generic interpretation approach which interprets 3D clothes configurations from the generic landmarks. This proposed interpretation demonstrates its robustness in terms of the variations in clothing's deformable configurations. To be more specific, in Chapter 5, the 'interpretation' visual perception architecture is presented. This is an enriched representation incorporating surface shape, topologies, textures and generic landmarks. This proposed method is applied to recognise clothes categories from free-configurations. From the comparison experiments, the proposed interpretation approach outperforms the state-of-the art method [Ramisa et al., 2013]. The former is an enriched representation that fuses multiple visual features extracted on the generic surface and landmarks, while the latter is a local based representation densely extracted on clothes surface. This result demonstrates that the proposed approach achieves a greater degree of robustness to clothing's deformable form. Moreover, compared to previously reported RGB based representations [Willimon et al., 2013a, 2011b], a better clothes recognition performance is achieved by the proposed 3D based method with fewer training garments. Therefore, the 3D based clothes representation is demonstrated to be more robust than RGB based representations.

The third hypothesis of this thesis is:

- By employing multiple perception-manipulation cycles, both robotic perception and manipulation goals can be incrementally approached in the integrated autonomous robot system (this process can be non-monotonic).

This hypothesis has been validated in two aspects: visually-guided manipulation and interactive recognition. Firstly, as it is reported in Chapter 3 and Chapter 4, the proposed flattening approach follows a perception-manipulation routine, incrementally approaching the pre-defined flattening goal (the halting criterion). Secondly, as reported in Chapter 6, the interactive recognition approach employs perception-manipulation cycles, modifying the unknown garment to different configurations until the prediction confidence modelled by Gaussian Process reaches a pre-defined threshold. From these two aspects, the experimental results validate this hypothesis that the robot is able to approach the manipulation or perception goal incrementally through employing perception-manipulation cycles.

7.4 Limitations and Future Work

Deformable objects (e.g. rope, clothing) perception and manipulation is one of the most challenging topics in robotics because of the flexible physical attributes and the quasi-infinite configurations of highly deformable objects. This thesis advances the state-of-the-art of robotic clothes perception and manipulation in three aspects: visually-guided manipulations, clothes recognition and interactive perception (as concluded in Section 7.2). However, there still exist limitations in the proposed approaches, and several interesting problems haven not been sufficiently explored yet. More specifically, the limitations and the potential solutions of proposed visually-guided manipulations, clothes recognition and interactive perception approaches are discussed in Sections 7.4.1, 7.4.2 and 7.4.3, respectively. For future work, teaching the robot learn dynamic manipulation skills is illustrated in Section 7.4.4; the idea of adapting human knowledge to robots is presented in Section 7.4.5; the proposals of error recovery and multiple robot collaboration are detailed in Section 7.4.6 and Section 7.4.7.

7.4.1 Visually-Guided Manipulation

In Chapter 3 and Chapter 4, the proposed visually-guided manipulation skills are capable of flattening different types of garments using generic visual perception and flattening strategies. However, there are two limitations that can be mitigated in future work. Firstly, the extra-class dissimilarity between different types of garments (their materials) and the intra-class dissimilarity between different garment instances are not considered. As a consequence, the spring parameter for calculating the flattening distance needs to be optimised manually for each instance. Future work of the reported visually-guided garment flattening proposes to investigate the potential solution of adapting reinforcement learning to learn the optimal spring parameter from the 'flatness' feedback during manipulation. Moreover, the proposed heuristic-based flattening is a greedy strategy that eliminates the largest wrinkle in each iteration. Future work has the potential to devise a global optimisation strategy or even a long-term goal optimisation strategy in order to achieve more effective and intelligent flattening.

7.4.2 Clothes Recognition

In Chapter 5, the proposed recognition pipeline based on an enriched representation from the 'interpretation' architecture is able to recognise various free-configurations of unknown clothes on the table. For the current work, only five categories of clothes are investigated. The future work should bridge the gap between this prototype approach and the industrial-level application, towards a more generalised and comprehensive recognition system, by

including more training clothes, and at least three new categories i.e. pants, dresses, hoodies. Also future research should aim for a faster prediction using paralleling computation for visual features, and sparse GP classification for large-scale data.

Moreover, the future work proposes to investigate the means of identifying physical attributes from proposed visual interpretation, in which the clothing's physical attributes (i.e. spring parameters) can be measured manually as the training examples and regression approaches can be used to estimated these physical attributes parameters from visual perceptions. Apart from the discriminative supervised learning used in the recognition pipeline, e.g. GP and SVM, active learning and on-line learning are proposed to be included in further work. More specifically, the robot should be able to interact with humans that actively ask questions when the prediction is not sufficiently confident and re-train the classifier after acquiring human instruction. Additionally, clothing instance identification based on colors, sizes, patterns, etc., are proposed to be investigated in the future work as a compliment to the proposed category recognition.

7.4.3 Interactive Perception

In order to improve the overall performance of the proposed interactive clothing recognition system, the future work proposes to investigate potential solutions for improving the robot's manipulation skills by including different types of manipulation, e.g. two-handed flattening or turning the garment inside-out. In the proposed interactive sorting method, an arbitrary manipulation strategy (action) is used in each iteration which is not sufficiently effective for some configurations. The future work proposes to investigate the relationship between actions' utilities and clothing's configurations. More specifically, the utility of a specific manipulation on a specific configuration can be measured by the increasing value of perception confidences between the current state and the consequent state. This kind of manipulation experience for training can be obtained in the form of $< state_t, action, state_{t+1}, untility >$ quadruplets by manually applying actions on garment. The manipulation utility of each action on a specific clothing's configuration has the potential to be predicted by regression techniques, and as a consequence, the best action can be chosen.

7.4.4 Learning Dynamic Manipulation Skills

Currently, heuristic manipulation strategies and discontinuous manipulation skills (not dynamics) are used in grasping, unfolding and folding. The future work would let the robot learn dynamic manipulation skills (the sequence of 6 Dimensional pose trajectories) in highly-skilled tasks (such as clothes folding and flattening in the air, etc.) through animation learning and reinforcement learning. By this means, the robot will be able to gain the knowledge

from manipulation experience and tune the dynamic manipulation trajectories depending on the physical attributes of clothing.

7.4.5 Adapting Human Knowledge to Robots

Supervised learning has been widely used on robot vision in the past decade. For the visual recognition phase, human knowledge (annotated data) is exactly the model that robots should learn. But for the visually-guided clothes manipulation, human-annotated data (e.g. for grasping position) may not be directly applicable to robots because of the constraints of clothes material (fabric), gripper configuration and so forth. The future work of this thesis proposes to translate human knowledge to the robot using active learning and on-line learning. For a visually-guided grasping task, provided the grasping position detection model has been learnt from human annotated data, the robot is supposed to actively give trails to explore the grasping uncertainties and refine the model after obtaining the tactile feedbacks. As a consequence, the robot would be able to adapt the models trained from annotated data to real clothes grasping scenarios, and also adapt the manipulation knowledge from grasping known material to grasping unknown materials.

7.4.6 Error Recovery

Within the state-of-the-art of clothes perception and manipulation, the whole pipeline of autonomous laundering has been achieved. However, error/accident detection and recovery have not been addressed yet. For example, in the sorting and unfolding tasks, the clothing may slip from the gripper and fall on the floor. In some cases of the folding task, the folding is completed unsuccessfully or aborted halfway. The future work proposes to explore the potential solutions of tracking the result of each step of folding, planing the decisions and recovering from faults.

7.4.7 Multiple Robot Collaboration

Multiple robot collaboration is able to solve complex laundering tasks more effectively. In future work, a multiple-agent autonomous laundering scenario has the potential to be achieved, in which the robots are able to separate, recognise, sort, unfold and fold garments collaboratively. Take University of Glasgow's scenarios for an example: the CloPeMa robot is responsible for manipulating garments, and a mobile Baxter robot is responsible for supervising the progress and moving the clothes buckets and piles at the proper times. Moreover, multi-agent collaborations have the potential to be adapted to large garment manipulation tasks, such as flattening or folding a sheet.

Appendix A

Figures and B-Spline Surface Fitting

Here the details of piece-wise B-Spline surface fitting and patches integration are presented. In this proposed approach, the point cloud is first divided into square patches on the x-y plane. An open uniform B-Spline surface is fitted to each patch [Koenderink and van Doorn, 1992]. Adjacent patches are connected together by blending the control points in order to ensure continuity. More specifically, for each $r \times s$ patch in the range map, the X, Y coordinates and depth value P of each pixel $X\{x_1, ..., x_r\}$,$Y\{y_1, ..., y_s\}$, $P = \{P_{(x_1,y_1)}, ..., P_{(x_i,y_j)}, ..., P_{(x_r,y_s)}\}$ can be obtained, and these points are then mapped into a two-parameter planar surface space uw (where u is with respect to x while w is with respect to y), then a Cartesian product B-Spline surface is therefore formulated as follows:

$$P(x_i, y_i) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} \Omega_{i,j} \alpha_{i,k}(x_i) \beta_{j,l}(y_i), \tag{A.1}$$

where $\Omega_{i,j}$ represents the control point situated at row i and column j. $\alpha_{i,k}(x)$ and $\beta_{j,l}(y)$ are the bases functions in the x and y directions, which can be calculated by [Rogers, 2001]:

$$\alpha(\beta)_{i,1}(t) = \begin{cases} 1, & if\ \overrightarrow{x}_i \leq t \leq \overrightarrow{x_{i+1}} \\ 0, & otherwise \end{cases} \quad and \tag{A.2}$$

$$\alpha(\beta)_{i,k}(t) = \frac{(t - \overrightarrow{x}_i)\alpha(\beta)_{i,k-1}(t)}{\overrightarrow{x}_{i+k-1} - \overrightarrow{x}_i} + \frac{(\overrightarrow{x}_{i+k} - t)\alpha(\beta)_{i+1,k-1}(t)}{\overrightarrow{x}_{i+k} - \overrightarrow{x}_{i+1}}, \tag{A.3}$$

where $\overrightarrow{x}_i$ is the ith element of the knot vector $\overrightarrow{x}$ and t is the parameter space. Eq.5.4 can thus be written as matrix form:

$$[P] = [\Phi][\Omega], \tag{A.4}$$

where $\Phi i, j = \alpha_{i,k}\beta_{j,l}$, while P is a $r \cdot s \times 3$ matrix containing the 3D coordinates of the range

map points, Φ is a $r \cdot s \times n \cdot m$ basis function matrix containing all the products of $\alpha_{i,k}(x)$ and $\beta_{j,l}(y)$, and Ω is a $n \cdot m \times 3$ matrix of control points coordinates. Thus, the B-Spline surface approximation is obtained by solving Eq. A.4 as a least-square problem:

$$[\Omega] = [[\Phi]^T[\Phi]]^{-1}[\Phi]^T[P]. \tag{A.5}$$

Finally, the new fitted points can be obtained by Equation A.4.

At this point, the range map surface is fitted separately in square patches. In order to get a continuous surface over the range map, a 3rd order uniform open knot vector $[0\,0\,0\,0\,1\,2\,2\,2\,2]$ is used to compute the basis function. Each patch is controlled by 5×5 control points in order to ensure C2 continuity. At the next step, control points are adjusted on the boundaries to achieve C1 continuity between adjacent patches.

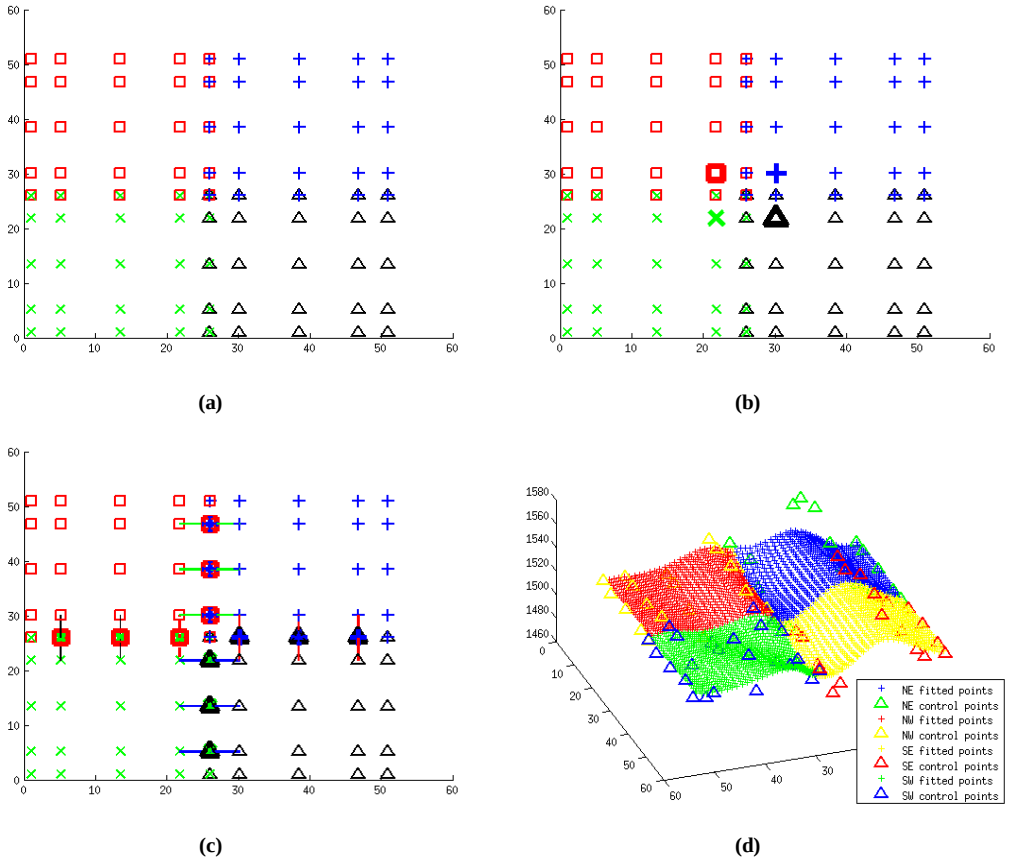


Figure A.1: The procedures of piecewise B-Spline surface fitting. In (a), (b) and (c), the distribution of control points are shown in the x-y plane. (a) Achieve C0 continuity by coinciding boundary control points. (b) Adjust the four inferior points (in bold face). (c) Enforce the control points in the boundary as the midpoint in horizontal and vertical directions. (d) Join result with C1 continuity.

The patch connecting process has three steps (as illustrated in Figure A.1). The first step is achieving C0 continuity on the patch boundaries by sharing control points (shown in Fig.A.1-

a). The control points on the boundaries are blended by setting them to the average of the boundary points from adjacent patches. Where more than two patches join (e.g. at patch corners) the average is taken over all adjacent boundary points. In the next step, the control points on the patch boundary always coincide with those in the adjacent patch. At the next step, the cross boundary derivatives (twist vector, t_{CP} where CP refers to control point) are enforced to be C1 continuous by averaging the twist vectors. The twist vectors in each patch are firstly calculated separately by Eq.A.6, before the four interior points (shown in Fig.A.1-b) are modified depending on the average of the twist vectors in Eq.A.7. A, B, C, D (shown as red, blue, green, black in Fig.A.1-b). The 5×5 control points grids of patch North West, North East, South West, South East are:

$$\begin{aligned} \tau_A &= A_{5,5} - A_{4,5} - A_{5,4} + A_{4,4} \\ \tau_B &= B_{5,2} - B_{4,2} - B_{5,1} + B_{4,1} \\ \tau_C &= C_{2,5} - C_{1,5} - C_{2,4} + C_{1,4} \\ \tau_D &= D_{2,2} - D_{1,2} - D_{2,1} + D_{1,1} \end{aligned} \tag{A.6}$$

$$\begin{aligned} A_{4,4} &= \tau - A_{5,5} + A_{4,5} + A_{5,4} \\ B_{4,2} &= -\tau + B_{5,2} - B_{5,1} + B_{4,1} \\ C_{2,4} &= -\tau + C_{2,5} - C_{1,5} + C_{1,4} \\ D_{2,2} &= \tau + D_{1,2} + D_{2,1} - D_{1,1}, \end{aligned} \tag{A.7}$$

where $A_{i,j}$, $B_{i,j}$, $C_{i,j}$, $D_{i,j}$ are control points in the ith row and the jth column of A, B, C and D, respectively, and $\tau = \frac{1}{4}(\tau_A + \tau_B + \tau_C + \tau_D)$. In the final step, the boundary control points are made C1 continuous along the vertical and horizontal directions. Specifically, as shown in Fig. A.1-c, the control points on the boundary are set as the average of its two neighbours in the horizontal and vertical directions (the midpoints in the lines in the figure). Moreover, in the four patches' center, the control point $A_{5,5}$ (also holds true for $B_{5,1}$, $C_{1,5}$, $D_{1,1}$) is set as the average of its four neighbours.

In Fig.A.1-d, a C1 continuous surface among four B-Spline patches is illustrated. In the implementation of this work, multiple spline fittings with different patches divisions are applied and the results are merged. Since each independent fitting is at least C1 continuous in every position, the surface obtained through averaging is also at least C1 continuous.

Having the B-Spline surface fitted, the fist order derivatives f_x, f_y and second order derivatives $f_x x$, $f_y y$, $f_x y$ at a surface point $P(x_i, y_i)$ can be calculated as follows:

$$f_x^p = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} \Omega_{i,j} \alpha'_{i,k}(x_i) \beta_{j,l}(y_i), \tag{A.8}$$

$$f_y^p = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} \Omega_{i,j}\alpha_{i,k}(x_i)\beta_{j,l}(y_i)', \tag{A.9}$$

$$f_{xx}^p = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} \Omega_{i,j}\alpha_{i,k}''(x_i)\beta_{j,l}(y_i), \tag{A.10}$$

$$f_{yy}^p = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} \Omega_{i,j}\alpha_{i,k}(x_i)\beta_{j,l}(y_i)'', \tag{A.11}$$

$$f_{xy}^p = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} \Omega_{i,j}\alpha_{i,k'}(x_i)\beta_{j,l}(y_i)', \tag{A.12}$$

Here, $\alpha_{i,k'^{(\prime)}}$ and $\beta_{j,l}(y_i)'^{(\prime)}$ are the derivatives of base functions, and more details of calculating them can be found in [Rogers, 2001].

Appendix B

Laplace Approximation for Multi-Class GP Classification

Following Eq. 6.4, from Bayes's rule, the posterior over latent variables can be inferred by:

$$\begin{aligned} p(f|X,y) &= p(y|f)p(f|X)/p(y|X) \\ &\propto p(y|f)p(f|X) \end{aligned} \tag{B.1}$$

Writing into log format, the log posterior can be obtained:

$$\Psi(f) = \log p(f|X,y) \propto \log p(f|X) + \log p(y|f), \tag{B.2}$$

where the prior of latent variable is a Gaussian $f|X \sim \mathcal{N}(0,K)$:

$$\log p(f|X) = -\frac{1}{2} f^T K^{-1} f - \frac{1}{2}\log|K| - \frac{Cn}{2}\log 2\pi, \tag{B.3}$$

and $p(y|f)$ is modelled by the soft-max function:

$$p(y_i^c|f_i) = \pi_i^c = \exp(f_i^c)/\sum_{c'=1}^{C}\exp(f_i^{c'}). \tag{B.4}$$

In Laplace approximation, the first order differential of log posterior $p(f|X,y)$ is computed:

$$\begin{aligned} \nabla \log p(f|X,y) &\triangleq \nabla \log p(f|X) + \nabla \log p(y|f) \\ &= -K^{-1}f + y - \pi \end{aligned} \tag{B.5}$$

where $\nabla \log p(f|X) = -K^{-1}f$ and $\nabla \log p(y|f) = y - \pi$. π is the vector with the length of Cn, containing soft-max probabilities of every latent variable π_i^c. Then, the second order

differential can be obtained by:

$$\nabla\nabla \log p(f|X,y) = -K^{-1} - W, \tag{B.6}$$

where W is a $Cn \times Cn$ matrix containing the $\frac{\partial^2}{\partial f_j^{c'} \partial f_k^{c''}} \log p(y_i^c|f_i)$, which can be calculated by:

$$\frac{\partial^2}{\partial f_j^{c'} \partial f_k^{c''}} \log p(y_j^{c'}|f_j) = \begin{cases} \pi_j^{c'} - \pi_j^{c'}\pi_k^{c''}, & if\ j = k,\ c' = c'' \\ -\pi_j^{c'}\pi_k^{c''}, & if\ j = k,\ c' \neq c'' \\ 0, & otherwise \end{cases}, \tag{B.7}$$

In the implementation of this work, W can be obtained by calculating $diag(\pi) - \Pi\Pi^T$, in which Π is obtained by vertically stacking diagonal matrices of $diag(\pi^c)$, and π^c is a sub-vector of π w.r.t category c. After the first and second order differentials are computed, the Newtown's method is applied to find the maximum of latent variable:

$$f^{new} = (K^{-1} + W)^{-1}(Wf + y - \pi). \tag{B.8}$$

Appendix C

Hyper-Parameters Optimisation for Gaussian Process Classification

From Laplace Approximation, the second order Taylor expansion of the posterior $p(f|X,y)$ is:

$$\Psi(f) \approx \Psi(\hat{f}) + \frac{1}{2}(f-\hat{f})^T \nabla\Psi(\hat{f}) + \frac{1}{2}(f-\hat{f})^T \nabla\nabla\Psi(\hat{f})(f-\hat{f}), \tag{C.1}$$

where $\nabla\Psi(\hat{f})$ is zero. Then, substituting approximated $\nabla\nabla\Psi(\hat{f})$ (calculated by Eq. B.6) into the marginal likelihood, the Laplace approximation of marginal likelihood is obtained:

$$\begin{aligned} p(y|X,\theta) &= \int p(y|f)p(f|X,\theta)df = \int \exp(\Psi(f))df \\ &= \exp(\Psi(\hat{f})) \int \exp(-\frac{1}{2}(f-\hat{f})^T(K^{-1}+W)(f-\hat{f}))df \end{aligned} \tag{C.2}$$

The Gaussian integral can be solved analytically, then the log marginal likelihood can be conducted as in [Rasmussen, 2006]:

$$\begin{aligned} \log q(y|X,\theta) \simeq -\frac{1}{2}\hat{f}^T K^{-1}\hat{f} + y^T\hat{f} - \sum_{i=1}^{n}\log(\sum_{c=1}^{C} exp\hat{f}_i^c) \\ -\frac{1}{2}\log|I_{Cn} + W^{\frac{1}{2}}KW^{\frac{1}{2}}| \end{aligned} \tag{C.3}$$

In Eq. C.3, since $\hat{f}$ and W have implicit relationship with hyper-parameters θ, the partial derivative of $\log\ q(y|X,\theta)$ w.r.t. θ can be computed into explicit and implicit parts.

$$\frac{\partial \log q(y|X,\theta)}{\partial\theta_j} \simeq \frac{\partial \log q(y|X,\theta)}{\partial\theta_j}|_{explict} + \sum_{i=1}^{Cn}\frac{\partial \log q(y|X,\theta)}{\partial\hat{f}_i^c}\frac{\partial\hat{f}}{\partial\theta_j} \tag{C.4}$$

Then the explicit part can be solved by:

$$\begin{aligned}\frac{\partial \log q(y|X,\theta)}{\partial \theta_j}|_{explict} = \frac{1}{2}\hat{f}^T K^{-1}\frac{\partial K}{\partial \theta_j}K^{-1}\hat{f} \\ -\frac{1}{2}tr((W^{-1}+K)^{-1}\frac{\partial K}{\partial \theta_j})\end{aligned} \tag{C.5}$$

The second term of Eq. C.4 can be solved by:

$$\frac{\partial \log q(y|X,\theta)}{\partial \hat{f}_i^c} = -K\hat{f}_i^c + \frac{\partial \log\ p(y|\hat{f})}{\partial \hat{f}_i^c} - \frac{1}{2}\frac{\partial \log|B|}{\partial \hat{f}_i^c} \tag{C.6}$$

Then $\frac{\partial q(f|X,y)}{\partial f} = 0$ when $f = \hat{f}$ is utilised, hence $-K\hat{f}_i^c + \nabla \log\ p(y|\hat{f}_i^c) = 0$, yielding:

$$\begin{aligned}\frac{\partial \log q(y|X,\theta)}{\partial \hat{f}_i^c} &= -\frac{1}{2}\frac{\partial \log|B|}{\hat{f}_i^c} \\ &= -\frac{1}{2}tr((W^{-1}+K)^{-1}\frac{\partial W}{\partial \hat{f}_i^c})\end{aligned} \tag{C.7}$$

in which W is the $Cn \times Cn$ matrix calculated by Eq. B.7. Then each element of $W_{j,k}$ (in jth row and kth column) is differentiated w.r.t. a specific scalar f_i^c. The elements of $\frac{\partial W_{j,k}}{\partial \hat{f}_i^c}$ if $j = k = i$ can be calculated as follows:

$$\begin{cases} (1-2\pi_j^{c'})(\pi_j^{c'} - \pi_j^{c'}\pi_k^{c''}),\ if\ c' = c'' = c \\ (1-2\pi_j^{c'})(-\pi_j^{c'}\pi_i^c),\ if(c' = c'') \neq c \\ -((\pi_j^{c'} - (\pi_j^{c'})^2)\pi_k^{c''} + \pi_j^{c'}(-\pi_k^{c''}\pi_i^c),\ if\ , c' \neq c'', c = c' \\ -((-\pi_j^{c'}\pi_i^c)\pi k^{c''} + \pi_j^{c'}(\pi_k^{c''} - \pi_k^{c''}\pi_i^c),\ if\ c' \neq c'', c = c'' \\ -((-\pi_j^{c'}\pi_i^c)\pi k^{c''} + \pi_j^{c'}(-\pi_k^{c''}\pi_i^c),\ if\ c' \neq c'', c'' \neq c \end{cases}, \tag{C.8}$$

and the rest are zeros.

In Eq. B.5, $\nabla \log p(f|X,y)$ should be 0 when f is at the maximum point. As a result, $-K^{-1}\hat{f} + \nabla \log p(y|f) = 0$ is obtained, therefore yielding $\hat{f} = K(\nabla \log p(y|f))$.

$$\frac{\partial \hat{f}}{\partial \theta_j} = \frac{\partial K}{\partial \theta_j}\nabla logp(y|f) + K\frac{\nabla\ logp(y|f)}{\partial \hat{f}}\frac{\partial \hat{f}}{\partial \theta_j} \tag{C.9}$$

Substituting $\frac{\nabla \log p(y|f)}{\partial \hat{f}} = \nabla\nabla\ \log p(y|f) = W$, $\nabla \log p(y|f) = y - \pi$, and solving Eq. C.9, it is:

$$\frac{\partial \hat{f}}{\partial \theta_j} = (I + KW)^{-1}\frac{\partial K}{\partial \theta_j}(y - \pi) \tag{C.10}$$

Having obtained $\partial \log q(y|X,\theta)/\partial \hat{f}_i^c$ and $\partial \hat{f}/\partial \theta_j$ by Eq. C.6 and Eq. C.10 respectively and substituting them into Eq. C.4, the derivative of Laplace approximated distribution can be obtained.

Bibliography

Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. Ieee, 2012.

R. Arandjelović and A. Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

Henry Harlyn Baker. Depth from edge and intensity based stereo. Technical report, DTIC Document, 1982.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.

Alexander Belyaev and Elena Anoshkina. Detection of surface creases in range data. In *Mathematics of Surfaces XI*, pages 50–61. Springer, 2005.

James R Bergen, Patrick Anandan, Keith J Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *Computer Vision?ECCV'92*, pages 237–252. Springer, 1992.

Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996.

Kevin W Bowyer, Kyong Chang, and Patrick Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+ 2d face recognition. *Computer Vision and Image Understanding*, 101(1):1–15, 2006.

Michael Brady, Jean Ponce, Alan Yuille, and Haruo Asada. Describing surfaces. *Computer Vision, Graphics, and Image Processing*, 32(1):1–28, 1985.

Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *ACM Transactions on Graphics (ToG)*, volume 21, pages 594–603. ACM, 2002.

Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5): 1055–1064, 1999.

Ondřej Chum and Andrew Zisserman. An exemplar model for learning object classes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

Paul Cockshott, Susanne Oehler, Tian Xu, Paul Siebert, and Gerardo Aragon-Camarasa. A parallel stereo vision algorithm. In *Many-Core Applications Research Community Symposium 2012*, 2012a.

WP Cockshott, Susanne Oehler, G Aragon Camarasa, J Siebert, and T Xu. A parallel stereo vision algorithm. 2012b.

Thomas M Cover and Peter E Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

Marco Cusumano-Towner, Arjun Singh, Stephen Miller, James F. O'Brien, and Pieter Abbeel. Bringing clothing into desired configurations with limited perception. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*

2011, pages 1–8, May 2011. URL http://graphics.berkeley.edu/papers/CusumanoTowner-BCD-2011-05/.

Boguslaw Cyganek and J Paul Siebert. *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2011.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.

Yining Deng and BS Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23 (8):800–810, 2001.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

A. Doumanoglou, A. Kargakos, Tae-Kyun Kim, and S. Malassiotis. Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 987–993, May 2014a. doi: 10.1109/ICRA.2014.6906974.

Andreas Doumanoglou, Tae-Kyun Kim, Xiaowei Zhao, and Sotiris Malassiotis. Active random forests: An application to autonomous unfolding of clothes. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 644–658. Springer International Publishing, 2014b. ISBN 978-3-319-10601-4. doi: 10.1007/978-3-319-10602-1_42. URL http://dx.doi.org/10.1007/978-3-319-10602-1_42.

Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013.

Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

Ken Goldberg et al. D-space and deform closure grasps of deformable parts. *The International Journal of Robotics Research*, 24(11):899–910, 2005.

G.G. Gordon. Face recognition based on depth and curvature features. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 808–810, Jun 1992. doi: 10.1109/CVPR.1992.223253.

Gösta H Granlund. Fourier preprocessing for hand print character recognition. *Computers, IEEE Transactions on*, 100(2):195–201, 1972.

Benjamin F Gregorski, Bernd Hamann, and Kenneth I Joy. Triangulation. In *Proc. ARPA Image Understanding Workshop*, pages 957–966, 1994.

Kyoko Hamajima and Masayoshi Kakikura. Planning strategy for task of unfolding clothes. *Robotics and Autonomous Systems*, 32(2):145–152, 2000.

Marsha J Hannah. Computer matching of areas in stereo images. Technical report, DTIC Document, 1974.

Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.

Heiko Hirschmüller and Daniel Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1582–1599, 2009.

Tin Kam Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, 1998.

A.M. Howard and George A. Bekey. Recursive learning for deformable object manipulation. In *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on*, pages 939–944, Jul 1997. doi: 10.1109/ICAR.1997.620294.

Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

Tommi Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.

Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, May 1999. ISSN 0162-8828. doi: 10.1109/34.765655.

Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. *ACM Transactions on Graphics (TOG)*, 26(3):24, 2007.

Robert Keys. Cubic convolution interpolation for digital image processing. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(6):1153–1160, 1981.

Alireza Khotanzad and Yaw Hua Hong. Invariant image recognition by zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):489–497, 1990.

Yasuyo Kita and Nobuyuki Kita. A model-driven method of estimating the state of clothes for manipulating it. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 63–69. IEEE, 2002.

Yasuyo Kita, Fuminori Saito, and Nobuyuki Kita. A deformable model driven visual method for handling clothes. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3889–3895. IEEE, 2004.

Yasuyo Kita, Toshio Ueshiba, Ee Sian Neo, and Nobuyuki Kita. Clothes state recognition using 3d observed data. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1220–1225. IEEE, 2009a.

Yasuyo Kita, Toshio Ueshiba, Ee Sian Neo, and Nobuyuki Kita. A method for handling a specific part of clothing by dual arms. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4180–4185. IEEE, 2009b.

Jan J Koenderink and Andrea J van Doorn. Surface shape and curvature scales. *Image and vision computing*, 10(8):557–564, 1992.

Marek Kopicki, Renaud Detry, Maxime Adjigble, Rustam Stolkin, Ales Leonardis, and Jeremy L Wyatt. One-shot learning and generation of dexterous grasps for novel objects. *The International Journal of Robotics Research*, page 0278364915594244, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Frank P Kuhl and Charles R Giardina. Elliptic fourier features of a closed contour. *Computer graphics and image processing*, 18(3):236–258, 1982.

Louisa Lam, Seong-Whan Lee, and Ching Y Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 14(9): 869–885, 1992.

Jeff Lander. Devil in the blue-faceted dress: Real-time cloth animation. *Game Developer Magazine*, 21, 1999.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

Yinxiao Li, Chih-Fan Chen, and Peter K. Allen. Recognition of deformable object category and pose. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014a.

Yinxiao Li, Yan Wang, Michael Case, Shih-Fu Chang, and Peter K Allen. Real-time pose estimation of deformable objects using a volumetric approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1046–1052. IEEE, 2014b.

Yinxiao Li, Danfei Xu, Yonghao Yue, Yan Wang, Shih-Fu Chang, Eitan Grinspun, and Peter K. Allen. Regrasping and unfolding of garments using predictive thin shell modeling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015a.

Yinxiao Li, Yonghao Yue, Danfei Xu, Eitan Grinspun, and Peter K Allen. Folding deformable objects using predictive simulation and trajectory optimization. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6000–6006. IEEE, 2015b.

Tsz-Wai Rachel Lo and J Paul Siebert. Local feature extraction and matching on range images: 2.5 d sift. *Computer Vision and Image Understanding*, 113(12):1235–1250, 2009.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999a.

David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999b.

DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 0920-5691.

Christiane Luible. *Study of mechanical properties in the simulation of 3D garments*. PhD thesis, University of Geneva, 2008.

Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2308–2315. IEEE, 2010.

Subhrajyoti Maji and Jagannath Malik. Object detection using a max-margin hough transform. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1038–1045. IEEE, 2009.

David Marr and Tomaso Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London B: Biological Sciences*, 204(1156):301–328, 1979.

David Marr and A Vision. A computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company*, 1982.

Stephen Miller, Mario Fritz, Trevor Darrell, and Pieter Abbeel. Parametrized shape models for clothing. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4861–4868. IEEE, 2011.

Stephen Miller, Jur Van Den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. A geometric approach to robotic laundry folding. *The International Journal of Robotics Research*, 31(2):249–267, 2012.

Richard A. Newcombe. Dense visual slam. In *PhD Thesis, Imperial College London*, 2012.

Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015.

Yuichi Ohta and Takeo Kanade. Stereo by intra-and inter-scanline search using dynamic programming. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2): 139–154, 1985.

Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, 23(3):609–612, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015768. URL `http://doi.acm.org/10.1145/1015706.1015768`.

T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, Jul 2002a. ISSN 0162-8828.

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002b.

Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995.

Lynn H Quam and Artificial Intelligence Center. Hierarchical warp stereo. *Readings in computer vision*, pages 80–86, 1984.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

Arnau Ramisa, Guillem Alenya, Francesc Moreno-Noguer, and Carme Torras. Using depth and appearance features for informed robot grasping of highly wrinkled clothes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1703–1708. IEEE, 2012.

Arnau Ramisa, Guillem Alenya, Francesc Moreno-Noguer, and Carme Torras. Finddd: A fast 3d descriptor to characterize textiles for robot manipulation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 824–830. IEEE, 2013.

Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42(3):287–320, 2001.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

David F Rogers. *An introduction to NURBS: with historical perspective*. Morgan Kaufmann, 2001.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 1–42, 2014.

Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Robot Manipulation*. Springer, 2013.

Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217, May 2009. doi: 10.1109/ROBOT.2009.5152473.

S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008a.

Ashutosh Saxena, Lawson LS Wong, and Andrew Y Ng. Learning grasp strategies with partial shape information. In *AAAI*, volume 3, pages 1491–1494, 2008b.

Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Learning 3-d object orientation from images. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 794–800. IEEE, 2009.

Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

John Schulman, Albert Lee, Jason Ho, and Pieter Abbeel. Tracking deformable objects with point clouds. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1130–1137. IEEE, 2013.

James Albert Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999.

David F Shanno. On broyden-fletcher-goldfarb-shanno method. *Journal of Optimization Theory and Applications*, 46(1):87–94, 1985.

Dave Shreiner, Bill The Khronos OpenGL ARB Working Group, et al. *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1*. Pearson Education, 2009.

JP Siebert and CW Urquhart. C3d: a novel vision-based 3-d data acquisition system. In *Image Processing for Broadcast and Video Production*, pages 170–180. Springer, 1995.

Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *British Machine Vision Conference*, 2013.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. *arXiv preprint arXiv:1511.02300*, 2015.

Shuran Song, Linguang Zhang, Jianxiong Xiao, Hedyeh Beyhaghi, Nishanth Dikkala, Eva Tardos, Y Métivier, JM Robson, A Zemmari, Zahra Aghazadeh, et al. Robot in a room: Toward perfect object recognition in closed environments. *arXiv preprint arXiv:1507.02703*, 2015.

Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Narf: 3d range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 44, 2010.

Jan Stria, Daniel Průša, and Václav Hlaváč. Polygonal models for clothing. In *Proc. Towards Autonomous Robotic System (TAROS)*, volume 8717 of *Lecture Notes in Artificial Intelligence*, pages 173–184. Springer, 9 2014a.

Jan Stria, Daniel Průša, Václav Hlaváč, Libor Wagner, Vladimír Petrík, Pavel Krsek, and Vladimír Smutný. Garment perception and its folding using a dual-arm robot. In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, pages 61–67. IEEE, 9 2014b.

Li Sun, Gerarado Aragon-Camarasa, Paul Cockshott, Simon Rogers, and J Paul. A heuristic-based approach for flattening wrinkled clothes. In *TAROS*, 2013.

Li Sun, Aragon-Camarasa Gerardo, Rogers Simon, and J. Paul Siebert. Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

Li Sun, Rogers Simon, Aragon-Camarasa Gerardo, and J. Paul Siebert. Recognising the clothing categories from free-configuration using gaussian-process-based interactive perception. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

Shuai Tang, Xiaoyu Wang, Xutao Lv, Tony X Han, James Keller, Zhihai He, Marjorie Skubic, and Shihong Lao. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Proceedings of 11th Asian Conference on Computer Vision (ACCV 2012)*, 2012.

Michal Jilich Thuy-Hong-Loan Le, Alberto Landini, Matteo Zoppi, Dimiter Zlatanov, and Rezia Molfino. On the development of a specialized flexible gripper for garment handling. *Journal of Automation and Control Engineering Vol*, 1(3), 2013.

Roger Y Tsai and Reimar K Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 554–561. IEEE, 1988.

Roger Y Tsai and Reimar K Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *Robotics and Automation, IEEE Transactions on*, 5(3): 345–358, 1989.

Raoul Tubiana, Jean-Michel Thomine, and Evelyn Mackin. *Examination of the hand and wrist*. CRC Press, 1998.

Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2): 154–171, 2013.

Jur Van Den Berg, Stephen Miller, Ken Goldberg, and Pieter Abbeel. Gravity-based robotic cloth folding. In *Algorithmic Foundations of Robotics IX*, pages 409–424. Springer, 2011.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

Christopher KI Williams and David Barber. Bayesian classification with gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1342–1351, 1998.

B. Willimon, I Walker, and S. Birchfield. A new approach to clothing classification using mid-level layers. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4271–4278, May 2013a. doi: 10.1109/ICRA.2013.6631181.

Bryan Willimon, Stan Birchfield, and Ian D Walker. Model for unfolding laundry using interactive perception. In *IROS*, pages 4871–4876, 2011a.

Bryan Willimon, S Birchfield, and Ian Walker. Classification of clothing using interactive perception. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1862–1868. IEEE, 2011b.

Bryan Willimon, Steven Hickson, Ian Walker, and Stan Birchfield. An energy minimization approach to 3d non-rigid deformable surface estimation using rgbd data. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2711–2717. IEEE, 2012.

Bryan Willimon, Ian Walker, and Stan Birchfield. 3d non-rigid deformable surface estimation without feature. 2013b.

Andrew Witkin, Demetri Terzopoulos, and Michael Kass. Signal matching through scale space. *International Journal of Computer Vision*, 1(2):133–144, 1987.

T Wong. Improvements to physically based cloth simulation. 2014.

Kimitoshi Yamazaki and Masayuki Inaba. A cloth detection method based on image wrinkle feature for daily assistive robots. In *MVA*, pages 366–369, 2009.

Jianchao Yang, Kai Yu, Yihong Gong, and Tingwen Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025. IEEE, 2011.

Jin Zhengping. *On the multi-scale iconic representation for low-level computer vision systems*. PhD thesis, PhD thesis, the Turing Institute and the University of Strathclyde, 1988.