Universidad Carlos III de Madrid
www.uc3m.es

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

TESIS DE MÁSTER

TOWARDS ROBOTIC CLOTHES FOLDING: A GARMENT-AGNOSTIC UNFOLDING ALGORITHM

Autor: David Estevez Fernandez
Tutor: Juan G. Victores
Director: Carlos Balaguer

MÁSTER OFICIAL EN
ROBÓTICA Y AUTOMATIZACIÓN

LEGANÉS, MADRID
MARZO 2016

UNIVERSIDAD CARLOS III DE MADRID
MÁSTER OFICIAL EN ROBÓTICA Y AUTOMATIZACIÓN

El tribunal aprueba la Tesis de Máster titulada "**Towards Robotic Clothes Folding:**
A Garment-Agnostic Unfolding Algorithm" realizada por **David Estevez Fernandez**.

Fecha: Marzo 2016

Tribunal:

Dr. Ramón Barber Castaño

Dr. Santiago Martínez de la Casa Díaz

Dr. Fernando Martín Monar

"It's better to have loved and lost than to have to do forty pounds of laundry a week."

Laurence J. Peter

Contents

Index of Figures ix

Index of Tables xiii

Acknowledgments xv

Abstract xvii

Resumen xix

1 Introduction 1

1.1 Objectives . . . 3

1.2 Document Structure . . . 4

2 State of the Art 7

2.1 Garment Modeling-Based Approaches . . . 8

2.2 Garment Manipulation-Based Approaches . . . 10

2.3 CloPeMa European Project . . . 14

3 A Garment-Agnostic Unfolding Algorithm 19

3.1 Garment Segmentation . . . 21

3.2 Garment Depth Map Clustering . . . 22

3.3 Garment Pick and Place Points 24

4 Garment Segmentation **27**

4.1 Background Subtraction 28

4.2 Garment Approximated Polygon Computation 31

5 Garment Depth Map Clustering **35**

5.1 Depth Image Preprocessing 36

5.2 Similar-Height Region Clustering 38

6 Garment Pick and Place Points **43**

6.1 Candidate Unfold Paths 43

6.2 Bumpiness 45

6.3 Pick and Place Points 48

7 Experiments and results **51**

7.1 Implementation details 51

7.2 Algorithm Evaluation 54

7.2.1 Garment Dataset 54

7.2.2 Experiments 55

7.3 Algorithm Validation 61

7.3.1 Experimental Setup 62

7.3.2 Experiments 63

8 Conclusions and future work **67**

8.1 Main contributions 67

8.2 Future Work 69

References **73**

Index of Figures

1.1 Current clothes folding solutions available for garment folding . 2

2.1 Basic action and model-driven strategy presented by Kita et al. . . 9
2.2 Models proposed by Miller et al. 9
2.3 Sample result from experiments performed by Li et al. 11
2.4 The complete pipeline of dexterous manipulation of deformable objects presented by Li et al. 12
2.5 PR2 robot during the manipulation of a pair of pants in order to reconfigure their pose. 13
2.6 CloPeMa FP7 European Project logo. 14
2.7 Block diagram of the method proposed by Mariolis et al. 15
2.8 Polygonal models proposed by Stria et al. 16
2.9 CloPeMa's bimanipulator robot manipulating a garment using the approach presented by Doumanoglou et al. 17

3.1 Block diagram showing the Garment Unfolding algorithm pipeline. 20
3.2 Two examples of background segmentation. 21
3.3 Several examples of Superpixel clustering algorithms. 23
3.4 An example of an unfolding operation, showing the humanoid robot and the unfolding trajectory to perform. 24

4.1 Block diagram showing an overview of the Garment Segmentation stage and its different steps. 27
4.2 Block diagram showing the different processes performed in the Garment Segmentation step. 29
4.3 Original RGB image captured by the robot sensor versus the binary segmentation mask obtained after the background subtraction process. 30
4.4 Block diagram showing the different processes performed in the Garment Approximated Polygon Calculation step. 31
4.5 Garment Contour, Garment Outline and Garment Appoximated Polygon. 33

5.1 Block diagram showing an overview of the Garment Depth Map Clustering stage and its different steps. 35
5.2 Depth image before and after preprocessing. 37
5.3 Example of application of the Watershed Tranform algorithm . . 39
5.4 The normalized depth images for different clothes are shown on the left side. 41

6.1 Block diagram showing an overview of the Garment Pick and Place Points stage and its different steps. 44
6.2 Initial candidate paths and valid set of paths. 45
6.3 Bumpiness along several candidate paths. 47
6.4 Final directions calculated for each garment provided to the system. 49

7.1 GUI developed in PySide to semi-automate the evaluation of the algorithm 53
7.2 Final output of the algorithm and the computed unfolding directions (Skirt and jacket) 56
7.3 Final output of the algorithm and the computed unfolding directions (Pants and Polo). 57

7.4 Final output of the algorithm and the computed unfolding directions (Robe and Hoodie). 58
7.5 Experimental setup used to test the algorithm 62
7.6 Comparison of the 3D point cloud with the Depth map obtained from its projection onto the table. 64
7.7 Garment unfolding operation being performed by the Teo humanoid robot 65

Index of Tables

7.1 Raw data results of the established scoring metric for each garment category and algorithm stage 59
7.2 Results analysis per stage and garment category, expressed in percentage (%) 60

Acknowledgments

I would like to thank Music for accompaning me along this journey we call life, and along the development of this thesis. I definitely wouldn't have done it without you. I encourage anyone reading this thesis to listen to some good music while doing so.

Thanks to my parents and brother for their unconditional love and support.

Thanks to Juan González-Gómez for encouraging me to finish this work, and to do so with the quality self-imposed by my perfectionist mind.

Thanks to Juan G. Víctores for mentoring me along the development of this work. Working with him has been to me a source of invaluable knowledge in science and life.

Thanks to my friends for their support and for those moments of fun and beers.

Thanks also to the people in the lab for ensuring the humanoid robot TEO was in perfect conditions for our tests with it.

Last but not least, I would like to thank the anonymous donor of the lost red hoodie used to conduct our first experiments. It was extremely useful to set the foundations of this work.

Abstract

Current approaches for robotic garment folding require a full view of an extended garment, in order to successfully apply a model-based folding sequence. In this thesis we present a garment-agnostic algorithm that requires no model to unfold clothes and works using a single view from an RGB-D sensor. Once the garment is unfolded, state of the art approaches for folding may be applied.

The algorithm presented is divided into 3 main stages. First, a Segmentation stage extracts the garment data from the background, and approximates its contour into a polygon. Then, a Clustering stage groups similar-height regions of the garment corresponding to different overlapped regions. Finally, a Pick and Place Points stage finds the most suitable points for grasping and releasing the garment for the unfolding process, based on a *bumpiness* value defined as the accumulated difference in height along selected candidate paths.

Experiments for evaluation of the algorithm have been performed over a dataset of 120 samples from a total of 6 different garment categories with one and two folds. Results have been analyzed, and present high scores for each of the stages that compose the algorithm. The unfolding algorithm also has been validated through experiments with a humanoid robot platform.

Resumen

En la actualidad, los métodos para el doblado automático de ropa usando robots requieren una vista completa de la prenda extendida, para su clasificación y posterior doblado basada en un modelo de la categoría a la que pertenece la prenda. En esta tesis, se presenta un algoritmo independiente del tipo de prenda que no requiere de un modelo previo para desdoblar ropa y que está basado en el uso de una única vista obtenida con un sensor RGB-D. Una vez desdoblada, se puede aplicar para su doblado cualquier algoritmo ya existente.

El algoritmo presentado en este trabajo está divido en 3 etapas principales. Primero, una Etapa de Segmentación separa la información de la prenda de la del fondo, y aproxima su contorno a un polígono. Después, una Etapa de Agrupación encuentra regiones de altura similar en la prenda, correspondientes a las distintas partes solapadas. Finalmente, una Etapa de Puntos de Agarre y Posicionamiento encuentra los puntos más adecuados para sujetar y soltar la prenda durante el proceso de desdoble, basados en un valor de *agrura*, definido como la diferencia de alturas acumulada a lo largo de las trayectorias de desdoble candidatas.

La evaluación del algoritmo se llevó a cabo a través de experimentos con un conjunto de datos que comprende 120 muestras de 6 categorías de prenda distintas, con uno y dos dobleces. Los resultados fueron analizados, y presenta

puntuaciones altas para cada una de las etapas que componen el algoritmo. El algoritmo de desdoble ha sido validado también a través de experimentos llevados a cabo con un robot humanoide.

Chapter 1

Introduction

Folding clothes is a common and necessary, but tedious, task for humans. Additionally, due to the increasing aging of the world population, a growing need exists for automated solutions to be able to help us with laundry. The textile industry another current source of demand for automated solutions. Current industrial solutions, shown in Figure 1.1 include human workers and voluminous machines. These machines are bulky, expensive, and require a large dedicated space, as they are intented to be used in an industrial environment. In addition, due to the complexity of garment manipulation tasks, these solutions are designed to be integrated into assembly lines, with several machines coupled to perform the whole process. Therefore, they are not suitable for domestic use. The proposed alternative is to use a robot to perform these garment manipulation tasks. A humanoid robot, designed to work in human environments, and to have human-like locomotion and manipulation capabilities seems to be a sensible choice.

Working with non-rigid objects such as clothes is a difficult task for robots, due to the complexity of modeling and manipulating deformable, thin objects. Clothes can be easily entangled when doing laundry, and recognizing individual garments and their category just from color or depth image analysis becomes

(a) *Human-based folding* **(b)** *Automatic folding*

Figure 1.1: *Current clothes folding solutions available for garment folding. On the left, human workers folding clothes in a textile factory*[1]*. On the right, an automated solution available in the market, manufactured by Texgraff©.*

an almost impossible task, due to occlusions amongst the cluttered clothes. Another challenging aspect when working with deformable objects is how to bring the object into a known configuration from an arbitrary initial state.

Extensive work can be found in literature about automated clothes folding once the garment category has been identified (that will be covered in the next chapter), as well as for modeling the garment for fold/wrinkle removal or selecting the most suitable grasping point/strategy. For that reason, this thesis focuses on how to unfold a clothing article that has been picked up from a pile of clothes and is placed on a flat surface. From that point, any of the existing approaches can be applied to fold the garment.

In this thesis the author presents a model-free garment-agnostic algorithm that can compute the pick and place points for a manipulator robot to iteratively unfold a garment, so that other algorithms can then determine the garment category and apply the folding sequence.

[1]Source: http://www.japantimes.co.jp/news/2012/12/19/reference/firms-move-some-eggs-out-of-china-basket/, last accessed: March 22, 2016.

It is assumed that a clothing article has already been separated from the rest of the clothes to be folded, and placed on a flat surface. The garment could have been placed on that surface either by a robot or by a human coworker, allowing a collaborative folding pipeline in which a human and a robot can perform different parts of the folding process. As the algorithm is not based on a geometrical model of the garment to be unfolded, it is general enough to be used with any category of garment, from towels and blankets to trousers or shirts, and with any number of folds. The presented approach consists in using a depth image from a single point of view to find regions of the garment overlapping other regions, which are considered to be folds. Then, all the possible candidate paths are studied to determine the unfolding direction. This is an iterative process to be repeated until the garment is fully unfolded.

1.1 Objectives

The main objective pursued in this work is to develop an algorithm that can estimate the grasping and release points for a deformable object so that a manipulator robot can iteratively unfold a garment. Once unfolded, its garment category and the folding sequence to apply can both be determined. From the aforehead mentioned algorithm we can deduce the following specific objectives:

- It should rely as little as possible in color or patterns present in the garment. This way, the algorithm is more robust and independent from the illumination conditions.
- It should provide a general method of detecting folds in deformable objects without a prior model of the garment to be unfolded. The absence of a prior model allows the algorithm to work with any kind of textile article, much as humans work with deformable objects.
- It should be able to estimate the best position of the grasping point, direction of movement, and release point in order to unfold the detected fold.

An incorrect pick and place sequence would cause the garment to be more entangled that in the initial state, making the unfolding task even more complicated.

This work will aim to accomplish these specific objectives in order to solve the presented main objective of garment unfolding.

1.2 Document Structure

This section presents an overview of the concepts and contents contained in each of the chapters of this thesis. This way, the reader may consume this work in a linear fashion, from start to end, or concentrate on specific parts that are relevant to him. For that purpose the author has tried, whenever possible, to write chapters as much self-contained as possible.

- **Chapter 2** provides an overview of the current state of the different methods and techniques to achieve automatic robot garment folding. It includes techniques based both on modeling and manipulation, as well as works performed as part of the CloPeMa FP7 European project. This chapter is recommended to obtain a general knowledge of other existing approaches that address the same challenges as this work.

- **Chapter 3** is related to the actual algorithm developed in this work. This chapter provides a general knowledge of the algorithm and its several stages, and therefore it is recommended that the reader reviews it to understand each concrete part in posterior chapters.

- **Chapter 4** deals with the different steps that constitute the Garment Segmentation stage. This stage is required to determine which pixels of the input images correspond to the garment and which ones correspond to the background.

- **Chapter 5** describes the different steps involved in the Garment Depth Map clustering stage. In this stage the pixels from the input depth image are grouped in regions of similar depth, to determine overlapping parts of the garment.

- **Chapter 6** presents the different steps involved in the Garment Pick and Place Points stage, explaining how the different steps in this stage generate pick and place points for the humanoid robot to manipulate the garment.

- **Chapter 7** is related to the different experiments performed to validate the algorithm. This chapter describes the experimental setup as well as the various experiments and results obtained from them.

- **Chapter 8** finally offers an analysis of the work and main contributions. The issues arised during the execution of this work are also mentioned, along with some lines of future work in order to address them.

Chapter 2

State of the Art

In this chapter, the current state of the art regarding garment folding is presented. Different approaches when working with garments can be found in the literature, and they will be described in detail.

Some approaches use 3D computer vision algorithms to create models of the garments or to fit captured gament data to a predefined model. These garment models are used then to apply folding algorithms or to select the best points for manipulation. Part of these models are borrowed from the Computer Graphics community, where garment representation and simulation has been widely studied to achieve realistic clothes behavior in Computer Graphics scenes. This will be covered in section 2.1.

In other approaches, the problem of garment folding is solved through robot interaction with the garment. By regrasping and changing the garment configuration, the garment category and current pose is detected, and the garment can be led to a target pose. These approaches will be seen in section 2.2.

Finally, the european project CloPeMa (Clothes Perception and Manipulation), devoted to perception and manipulation of fabric, textiles and garments, is described. Their research most relevant to this thesis is explained in the last section of this chapter, section 2.3.

2.1 Garment Modeling-Based Approaches

A significant amount of work conducted in this topic has been focused on modeling the different garment categories for both unfolded extended garments and for grasped garments. The computer graphics community has also contributed with extensive work on the specifics of clothes modeling due to their need for realistic representation of fabrics and garments. A review of these modeling methods can be found in (Qian, Yan, & Ai-fang, 2008).

Kita et al. propose a method that uses a deformable model to calculate the state of hanging clothes based on 3D observed data (Kita, Saito, & Kita, 2004; Kita, Ueshiba, Neo, & Kita, 2009). This calculation is performed by generating a set of candidate shapes predicted by physical simulations of hanging clothes and later comparison of them with the observed data. To fit the observed 3D data better, each generated shape is further deformed and the shape that is more consistent with the observed data is selected. This basic action and model-driven startegy presented by Kita et al. is depicted in Figure 2.1, in which the garment is held by a point with a robotic arm and different deformable models are tested against the garment data. The model that best explains the data is selected to calculate the second grasping point.

Miller et al. present an approach to modeling the clothes when already spread out on a flat surface in (Miller, Fritz, Darrell, & Abbeel, 2011). A series of parametrized shape models are proposed, each clothing category having its own model. Garment variability is solved through variation of those parameters. Once the garment has been modeled with their method, a preprogrammed folding sequence can be performed. Figure 2.2 shows two examples of the parameterized shape models proposed by Miller et al. The points shown in red are the skeletal points selected to fit the model to the garment data obtained by the vision system. Using those points and the parameters represented by the red segments the rest of the key points and dimensions are calculated to complete the model.

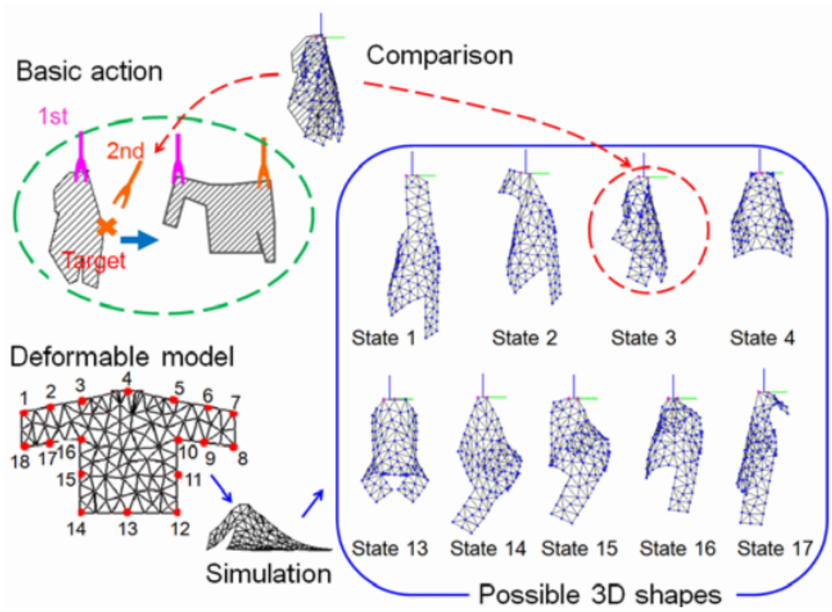


Figure 2.1: *Basic action and model-driven strategy presented in (Kita et al., 2009).*

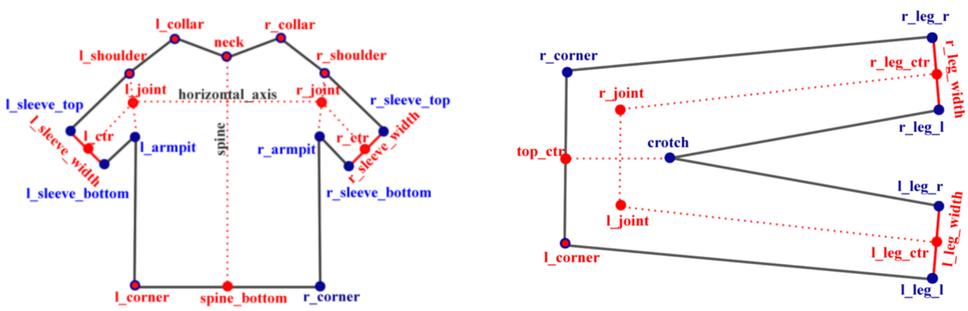


Figure 2.2: *Models proposed in (Miller et al., 2011).*

A method for classifying and estimating the poses of deformable objects is presented in (Li, Chen, & Allen, 2014). This method consists in creating a training set of deformable objects by off-line simulation of different garments, extracting depth images from different points of view. Then, a codebook is built for a set of different poses of each deformable object by extracting features from the dataset and applying sparse coding and dictionary learning. With this codebook, classifying deformable objects on different categories and estimating their current pose is possible, for later regrasping or folding the garment.

The previous method was improved in (Li, Wang, Case, Chang, & Allen, 2014), by extracting the features directly from the 3D data, dividing the hanging garment in different cells via layers, rings and sectors of the bounding cylinder. Each of the sectors becomes a binary feature, using the Signed Distance Function to check if the cell is inside the voxel where the center of the cell belongs, and is then arranged in a feature vector. A Hamming distance, whose weights are learned from the simulated dataset merged with some models reconstructed from real word Kinect point clouds, is used to estimate the object category and pose given an input reconstructed mesh model. A sample result from the experiments performed by Li et al. is demonstrated by Figure 2.3.

2.2 Garment Manipulation-Based Approaches

Clothing article manipulation is another field in which extensive work has been done. Based on the previous recognition algorithm, Li et al. present in (Li, Xu, et al., 2015) a method for unfolding deformable objects with a bi-manipulator robot. With this method, the robot is capable of taking a clothing article from an unknown state to a known state by iterative regrasping, detecting the most suitable grasping points in each state to achieve its goal. For locating the most suitable grasping points, the 3D point cloud obtained by the robot is matched to the mesh model, that incorporates the information about the best regions to grasp in order to unfold the garment.

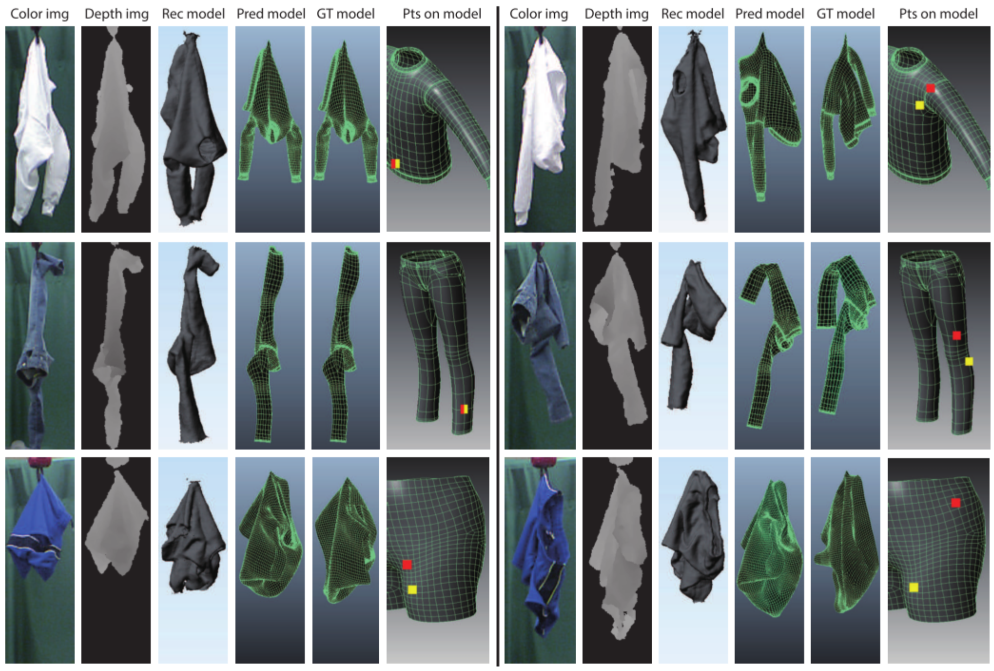


Figure 2.3: *Sample result from experiments performed in (Li, Wang, et al., 2014). Each row corresponds to a different garment, grouped into 2 different views of each garment (columns 1-6 and 7-12). The first and second pictures of each group correspond to color and depth images captured with the Kinect sensor. The third picture shows the reconstucted model for each garment. The remaining pictures show, from left to right, the predicted simulated model, the ground truth simulated model and the predicted and ground truth grasping points, overlaid over a 3D view of the unfolded garment.*

Figure 2.4 shows the complete pipeline of dexterous manipulation of deformable objects presented by Li et al. The pipeline of a robot folding a garment from a random state spans initial grasping, pose estimation, regrasping and placing the garment on a table. Actions indicated in the red rectangle, regrasp the object and repeat the pose estimation step, are performed by the the robot in case that the recognition is not successful.

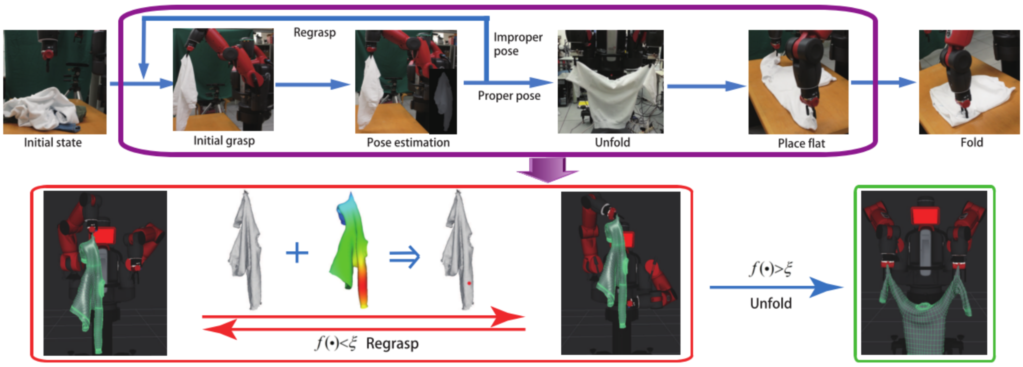


Figure 2.4: *The complete pipeline of dexterous manipulation of deformable objects presented in (Li, Xu, et al., 2015).*

The method introduced by Cusumano-Towner et al. in (Cusumano-Towner, Singh, Miller, O'Brien, & Abbeel, 2011) allows a bi-manipulator robot to identify a clothing article, estimate its current state and achieve a desired configuration, generalizing to previously unseen garments. For that purpose, the robot uses a Hidden Markov Model (HMM) throughout a sequence of manipulations and observations, in conjunction with a relaxation of a strain-limiting finite element model for cloth simulation that can be solved via convex optimization. Figure 2.5 depicts a PR2 robot performing a reconfiguration operation based on Cusumano-Tower et al.'s method.

Osawa et al. propose in (Osawa, Seki, & Kamiya, 2007) a method to unfold garments in order to classify them. It consists in alternatively regrasping clothing and expanding them using a two-arms manipulator. The garment is grasped

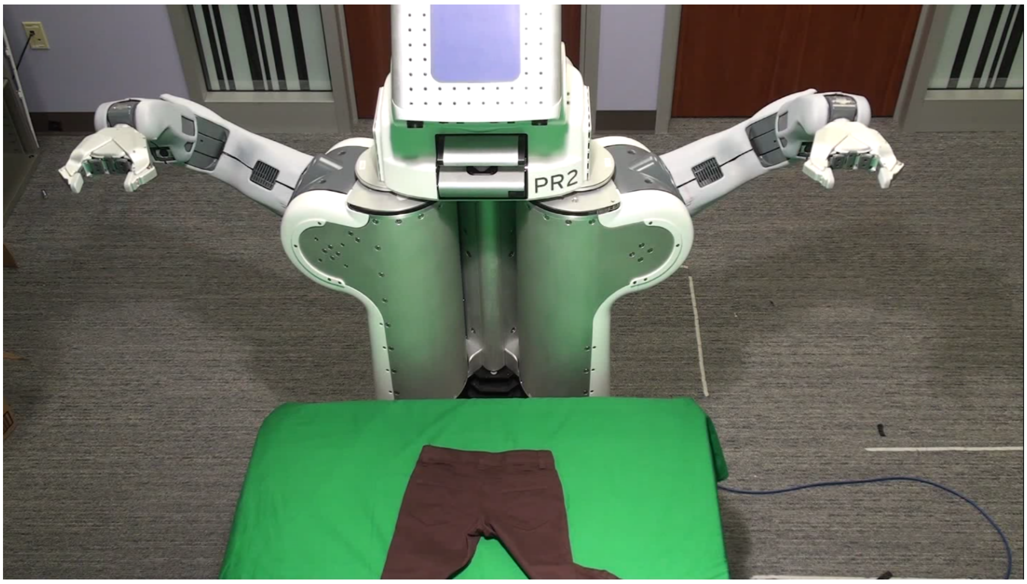


Figure 2.5: *PR2 robot in an intermediate state during the manipulation of a pair of pants in order to reconfigure their pose.*

with one arm and the lowest point is located by rotating the piece of clothing, which is used as a grasping point for the other arm. If the garment has any fold when extended, it is placed over a flat surface to repeat this process util the the garment is fully spread out.

To detect the best grasping points for a clothing article, Ramisa (Ramisa, Alenya, Moreno-Noguer, & Torras, 2012) performs the identification in a single step, even with highly wrinkled clothes. This detector is based in a Bag of Features detector, using as input a combination of appearance and 3D geometric features.

The most similar work we can find in the related literature is the method for unfolding clothes presented by Willimon et al. in (Willimon, Birchfield, & Walker, 2011). Their method, which also focuses in clothes unfolding prior to automated folding, uses several features obtained from a depth image, such as peak regions and corners location, to determine the location and orientation most suitable for interaction with the garment. Two main steps are performed: first, the clothing article is flattened using RGB information from the camera and, then, depth information is used to extract the features used to estimate how to unfold the garment.

2.3 CloPeMa European Project

CloPeMa[1] is a recent EU-FP7 research project (2012-2015) whose objective is to advance the state of the art in perception and manipulation of fabric, textiles and garments. The project's official logo can be seen in Figure 2.6.

Figure 2.6: *CloPeMa FP7 European Project logo.*

As part of the CloPeMa project, a method to detect single folds has been presented by Mariolis et al. in (Mariolis & Malassiotis, 2013, 2015). In order to detect such folds, first, a database of unfolded clothes templates is built. These templates are later used to perform a shape matching between the folded garment shape, obtained by the camera, and the unfolded garment model. This process is iterative, and the initial results are feedbacked to adapt the model for a better fit. A block diagram of this method is portrayed in Figure 2.7.

[1] http://www.clopema.eu/, last accessed: March 22, 2016.

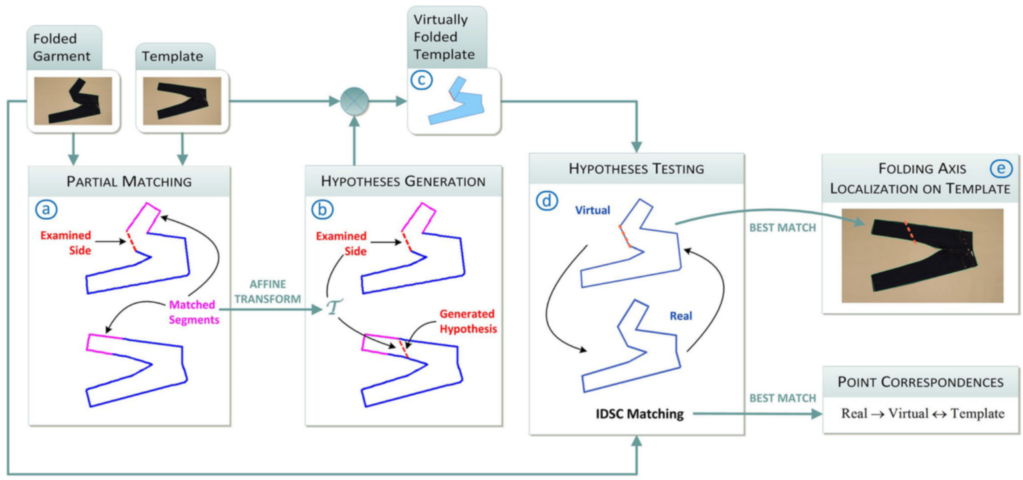
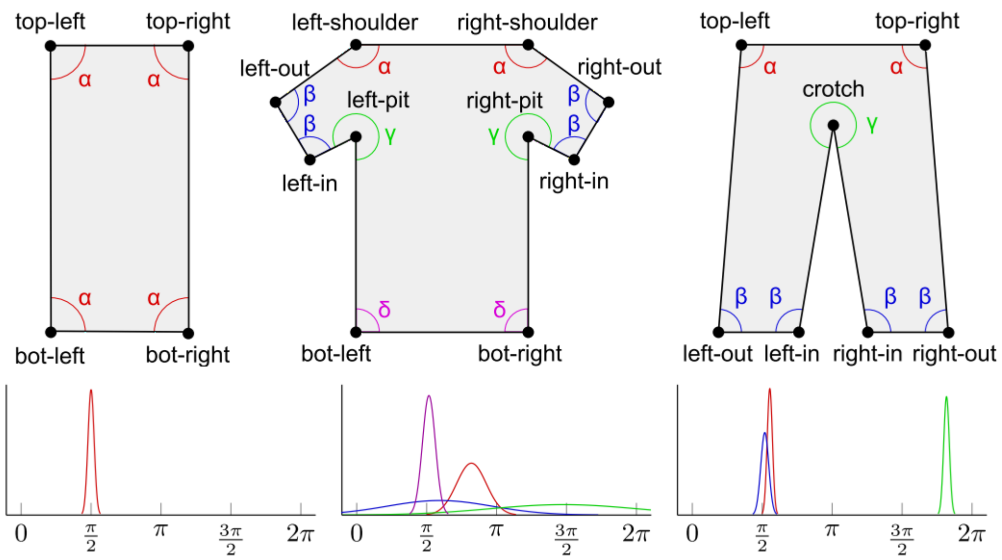


Figure 2.7: *Block diagram of the method proposed in (Mariolis & Malassiotis, 2015). This method performs shape matching of folded garments and unfolded templates. In the first stages, the garment and the unfolded template are matched and transformed to generate an hypothesis. An Inner Distance Shape Contexts (IDSC) algorithm is then applied to test the hypotheses and find the best match for the folding axis on the template.*

Stria et al. propose in (Stria, Průša, & Hlaváč, 2014; Stria, Průša, Hlaváč, Wagner, et al., 2014) a polygon-based model for clothes configuration recognition using the estimated position of the most important landmarks in the clothing article. Once identified, these landmarks can be used for automated folding using a robotic manipulator. The clothes contour is extracted from a RGB image and processed using a modified grabcut algorithm and dynamic programming methods are used to fit it to the polygonal model. Polygonal models proposed in (Stria, Průša, & Hlaváč, 2014) for several categories of clothes are shown in Figure 2.8. The upper figures show the different angular parameters that comprise the model of each garment category. Beneath each one of them, the circular distribution of each of the angles of the model occurring in real garments.

Doumanoglou et al. follow in (Doumanoglou, Kim, Zhao, & Malassiotis, 2014) an approach based on Active Random Forests to recognize clothing articles from depth images. This classifier allows the robot to perform actions to



(a) Towel (b) Short shirt (c) Pants

Figure 2.8: *Polygonal models proposed in (Stria, Průša, & Hlaváč, 2014) for several categories of clothes. Beneath each model the angular distribution for each angle is shown.*

collect extra information in order to disambiguate the current hypotheses, such as changing the viewpoint. In (Doumanoglou, Kargakos, Kim, & Malassiotis, 2014) they extend this approach to detect the optimal grasping points to unfold the garment. CloPeMa's bimanipulator robot manipulating a garment using the approach presented by Doumanoglou et al. is depicted in Figure 2.9. The robot is made of two industrial robotic MA1400 arms, used in the welding industry, and has a binocular system composed of two Nikon DSLR cameras (D5100). Other sensors, such as force or photometric close-range sensors are also integrated in the robot. Active Random Forests are applied by the robotic system to perform garment recognition, changing the garment pose if disambiguation is required.

Within the context of the CloPeMa project, some work on wrinkles removal has been made by Sun et al. (Sun, Aragon-Camarasa, Rogers, & Siebert, 2015) with a stereo vision system and a dual manipulator robot. Wrinkles are first

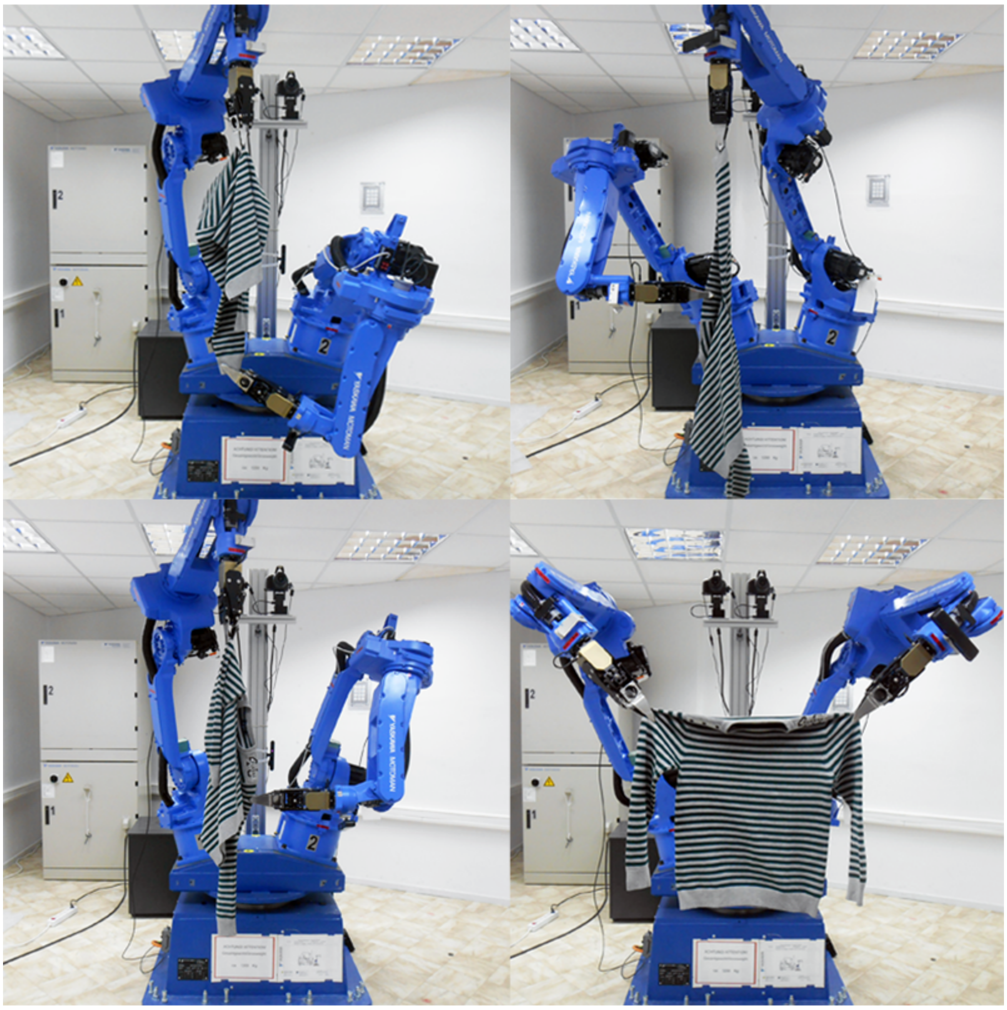


Figure 2.9: *CloPeMa's bimanipulator robot manipulating a garment using the approach presented in (Doumanoglou, Kim, et al., 2014).*

identified using a depth map of the cloth, which is later B-spline smoothed and parsed by shape and topology into different wrikle structures, that are later ranked by size and removed.

Chapter 3

A Garment-Agnostic Unfolding Algorithm

This chapter presents an overview of a garment-agnostic algorithm to unfold clothes using depth sensor data, that will be extended in later chapters. The pipeline for folding clothes followed throughout robotic literature is based on how it is performed by humans. This allows these tasks to be executed in the same environments as humans, aiming towards a fully-automated or collaborative fashion. Garments are extracted from the washing or drying machine forming a pile, and an iterative process begins. First, an individual clothing article is picked from the pile. Then, the garment is extended in the air or assisted by a flat surface, during which unfolding and wrinkle removing procedures may be performed for aiding the posterior classification and manipulation of the garment. A classification procedure is applied to fit clothing article within a certain garment category. Finally, a standard manipulation sequence specific to its category is applied to fold the garment for storage. This iterative process is repeated until there are no clothes left on the pile.

Our work differs with most of the state of the art in the fact that garment models are not used. Our approach uses depth information captured with an

RGB-D sensor to detect folds in a garment. The most suitable grasping position, unfolding direction, and release point are computed.

The algorithm can be divided into three main stages, shown in Figure 3.1. First, the background is extracted from the image and the contour of the garment is approximated. Then, a depth map clustering process is performed to estimate overlapping regions. Finally, the information obtained in the previous process is used to determine if there are any folds present and how to interact with the garment to unfold it. The whole unfolding pipeline is an iterative process, that should be repeated again until the garment is completely extended.

The algorithm has been open sourced and is available online[1].

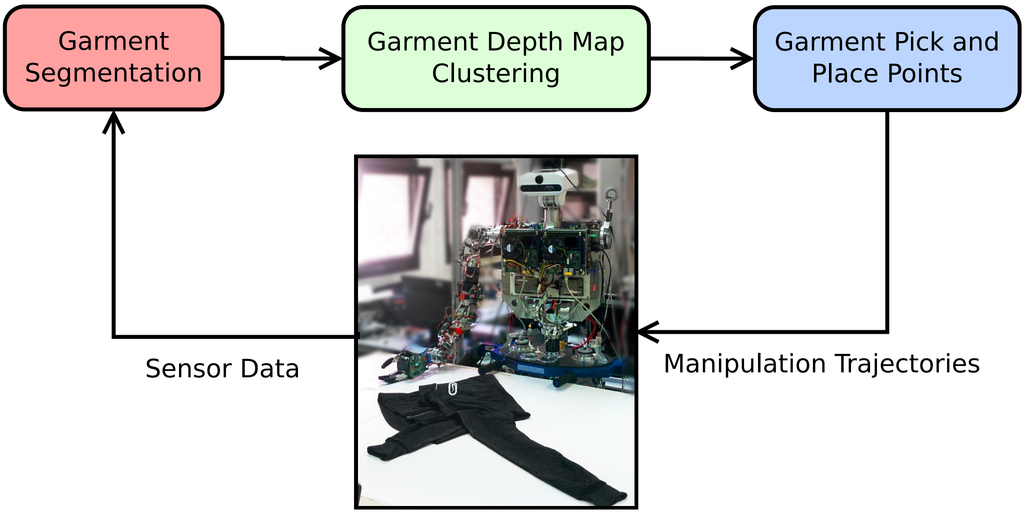


Figure 3.1: *Block diagram showing the Garment Unfolding algorithm pipeline. The pipeline uses RGB and depth images captured from the robot sensors as inputs and process them in 3 consecutive stages. This process is iterative, and it is applied until the garment is fully unfolded.*

[1]https://github.com/roboticslab-uc3m/textiles, last accessed: March 22, 2016.

3.1 Garment Segmentation

The objective of this garment segmentation stage is to substract the background from the input RGB-D image to obtain only the resulting garment data, as well as to extract the simplified garment contour. Figure 3.2 depicts a generic sample of this background subtraction and contour extraction process.

Figure 3.2: *Two examples of background segmentation. In the left image (Z. Ren & Shakhnarovich, 2013), the foreground objects contour is highlighted in red. In the right image (Nieuwenhuis & Cremers, 2013), the foreground object (the tiger) is highlighted in red, and the background is highlighted in green.*

Prior to any further analysis, a background subtraction step has to be performed in order to remove all information not related to the garment. Based on the assumption that the clothing article is placed on top of a flat white surface, the garment can be segmented by color. This can be performed by threshold operations on the red, green and blue channels of the RGB space, or on the hue, saturation and value channels of the HSV space. Other color spaces could also be used. GrabCut (Rother, Kolmogorov, & Blake, 2004), that uses gaussian mixture models (GMMs) and iterative energy minimization, is another available method to separate the background and foreground based on color.

Another potential approach is to remove the horizontal plane of the table using depth information. This approach can be further improved by combining the results obtained by the depth-based segmentation with a color segmentation, in case the garment color and the background are dissimilar enough.

Once the background is extracted, the contour of the garment has to be determined. Several methods exist for this purpose. The Topological Analysis by Border Following algorithm developed by Suzuki and Abe (Suzuki et al., 1985) is a widely used algorithm for connected-component labeling and countour finding. A Marching Squares algorithm, a variation of the Marching Cubes algorithm (Lorensen & Cline, 1987), can be also applied to find constant valued contours in an image.

After the contour is extracted, it is then approximated to a polygon, e.g. using the Ramer-Douglas-Peucker algorithm (Ramer, 1972; Douglas & Peucker, 1973). Alternative algorithms exist, such as the Teh-Chin algorithm to find Dominant Points (Teh & Chin, 1989). Each of the segments obtained is a candidate for being a fold axis.

The garment segmentation process used in this thesis is further explained in chapter 4.

3.2 Garment Depth Map Clustering

When the garment has been identified, garment regions with similar height points must be clustered and labeled. By using only depth information, as opposed to using RGB images, makes our algorithm independent of the colors and patterns present in the garments. Many algorithms exist to perform this clustering step.

For general purpose clustering over any set of unlabeled data we can find several algorithms that can be also applied to images. K-means (MacQueen et al., 1967), Mean Shift (Comaniciu & Meer, 2002) or DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) are some examples of clustering algorithms that can be applied to group pixels with similar depth.

A particular case of clustering algorithms are those which are applied to obtain superpixels. A superpixel (X. Ren & Malik, 2003) is a local region of the image with similar characteristics, such as color, texture, etc. Several examples of superpixel can be seen in Figure 3.3. This reduces the number of pixels to regions that are local and coherent, preserving most of the structure of the diferent items at the scale of interest. Many algorithms exist to perform superpixel clustering, such as SLIC (Simple Linear Iterative Clustering (Achanta et al., 2010), Quickshift (Vedaldi & Soatto, 2008), Turbopixels (Levinshtein et al., 2009) and Normalized Cuts (Mori, 2005).

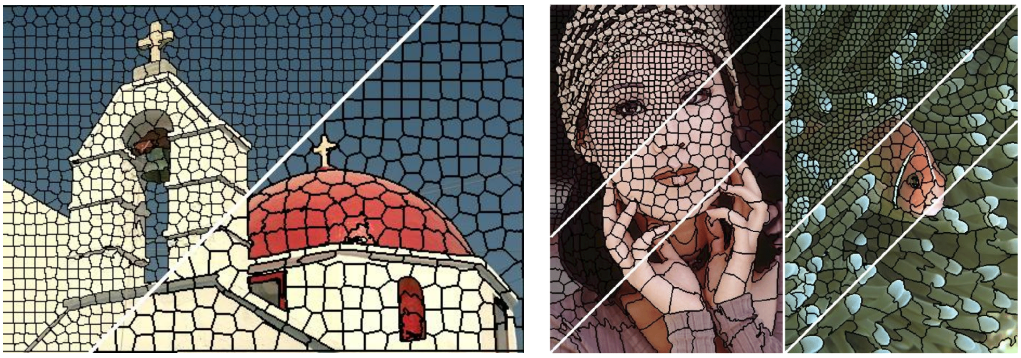


Figure 3.3: *Several examples of Superpixel clustering algorithms. The left image (Mori, 2005) corresponds to a Normalized Cut algorithm with two different average cluster sizes: 100 pixels in the case of the upper left part of the image, and 300 in the lower right part. The right image (Achanta et al., 2012) corresponds to a SLIC algorithm (Achanta et al., 2010) applied to obtain average sizes of 64, 256 and 1024 pixels.*

When working with images, the clustering process of finding regions both compact and with homogeneous characteristics is known as segmentation. Many image segmentation algorithms exist based on similar regions and closed contours. One of these algorithms, widely used, is the Watershed transform algorithm (Digabel & Lantuéjoul, 1978), which interprets the image as a topological surface, filling the image with water from several point sources until they merge, and creating walls in those intersection contours. The regions isolated by these walls are the segmented regions.

The garment depth map clustering process used in this thesis is developed in more detail in chapter 5.

3.3 Garment Pick and Place Points

Once we have clustered the depth image in regions of similar height, we can proceed to analyze those regions. The objective is to find which parts of the region outline correspond to the fold line, which is physically connected to the rest of the garment. Once the fold line has been detected, pick and place points have to be found for the robot to perform the unfolding operation. Figure 3.4 displays an example of this unfolding operation, with its pick and place points marked.

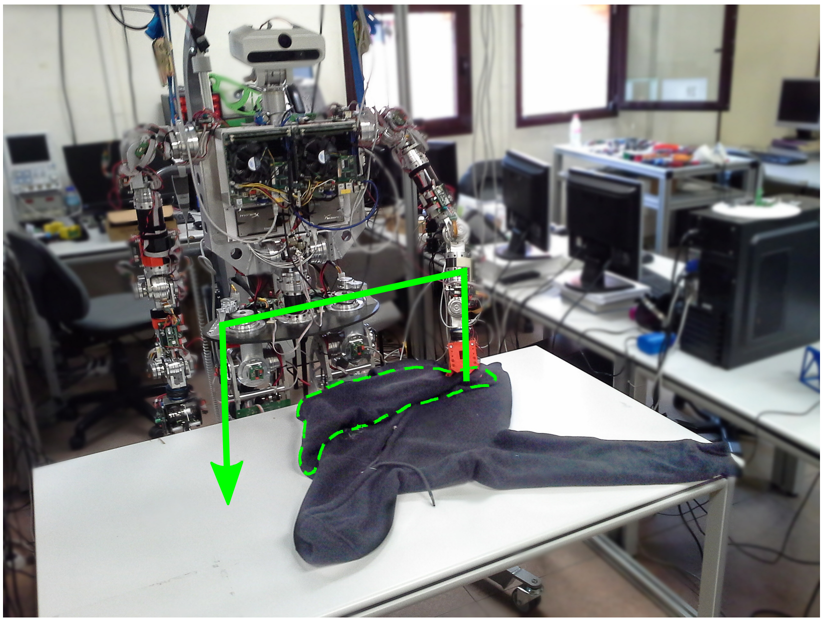


Figure 3.4: *An example of an unfolding operation, showing the humanoid robot and the unfolding trajectory to perform. The folded part that is being manipulated, the sleeve, is highlighted with a green dashed line. The green arrow line represents the unfolding trajectory, with the pick and place points at the beginning and the end of it, respectively.*

If grouping of certain segmented regions is required, i.e. there are small superpixels with similar characteristics, a merging algorithm has to be performed. A graph-based approach could be used for both merging clusters and determining which clusters are physically connected, either by the fold line or because they belong to the same garment region.

Once the overlapping fold is identified, and the fold line is known, we select pick and place points for performing the unfold operation. This operation, that will be executed by a robot, has to take the garment from the original state to a more "unfolded" state.

Several choices exist for pick points, as well as for place points. To pick the garment, one can assume that the highest points in the overlapped region are good candidates, as they probably are easier for the robot to grasp, and they will be separated from the underlying garment. Other potential choices are the points that belong to the border of the overlapping region, except for those which belong to the fold line. These approaches are typically followed by humans when performing unfolding tasks. For place points, the points should be far enough from the garment to completely unfold the overlapping region, but not far enough for the garment to be dragged with the movement. It is also convenient to choose points that lie on the opposite side of the folding line with respect to the pick point, otherwise is almost impossible to perform the unfold operation.

Chapter 4

Garment Segmentation

This chapter is devoted to the first stage of our algorithm, segmenting the garment data from the raw sensor data. The input data is an RGB image from the RGB-D data stream. Garment segmentation is perfomed over this image, and the garment contour is then extracted and simplified, ready to be used in posterior stages. The block diagram for this stage is shown in Figure 4.1.

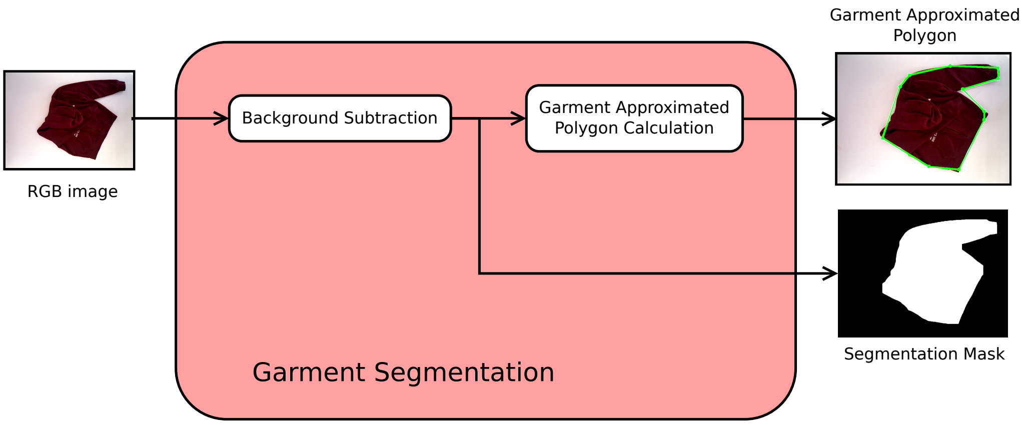


Figure 4.1: *Block diagram showing an overview of the Garment Segmentation stage and its different steps. The input of this stage is an RGB image of the garment, and the outputs are the segmentation mask and the garment approximated polygon. The segmentation mask is also used as a by-product to obtain the approximated polygon.*

4.1 Background Subtraction

The RGB image obtained from the robot sensor contains both the clothing article and the table on which it rests. Therefore, after retrieving the data, a background subtraction step is required to classify whether a pixel represents the garment or the table. Figure 4.2 shows an overview of the different processes involved in this step.

For this purpose, many methods could have been chosen, based on both color and depth information, as discussed in section 3.1. For instance, GrabCut (Rother et al., 2004) was discarded for background subtraction as it requires user input to select background and foreground samples. Furthermore, it is computationally expensive compared with simple thresholding methods. As the main focus of our work is unfolding clothes, a simple color-based method was selected instead.

We work under the assumption that the garment has been placed over a flat white surface, as opposed to the garment which is much more colorful (higher Saturation). Therefore, the RGB image is converted to the HSV space. Working in the HSV space gives us direct information about our magnitudes of interest: Saturation (S) and Value (V). We are not interested in detecting a particular color, but to detect a colored item, so HSV is a more sensible choice of color space than RGB.

A filtering process is added to increase robustness and reduce the effect of the noise on the background subtraction. This process is achieved using a convolution with a 5x5 and $\sigma = 1.1$ Gaussian kernel computed on the Saturation and Value channels of the HSV image.

Once the image is converted to the HSV color space and filtered, a thresholding operation is then applied to the filtered image, using Otsu's algorithm (Otsu, 1975) to obtain the optimal threshold values. Pixels with low amount of Saturation, and high Value are classified as being part of the table, as opposed to saturated or dark pixels.

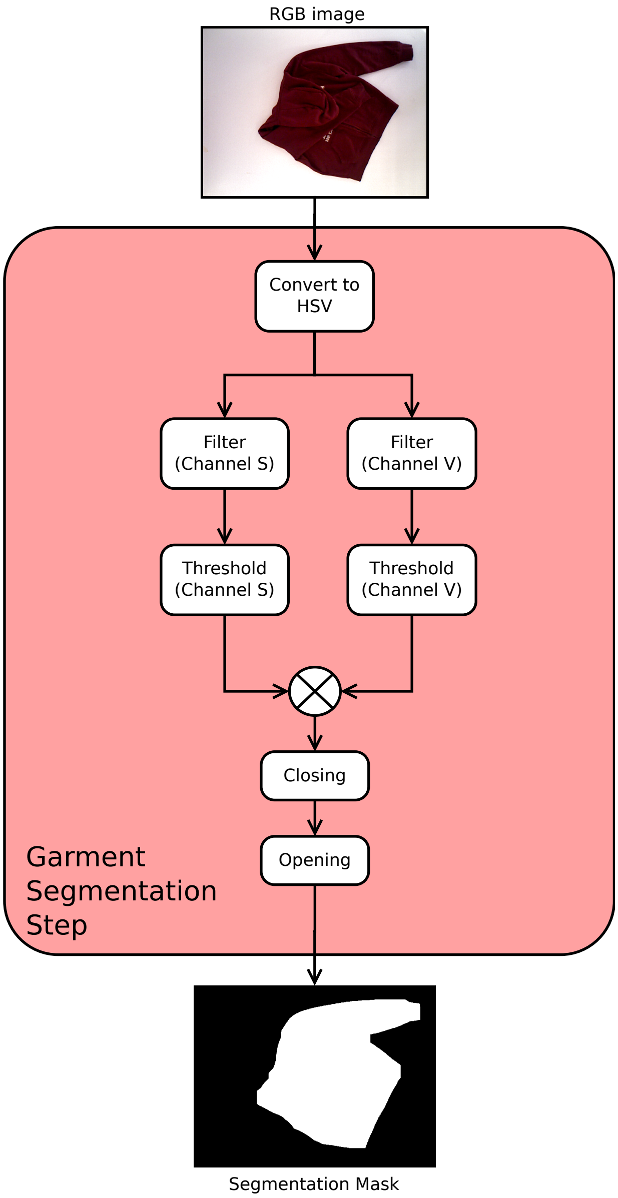


Figure 4.2: *Block diagram showing the different processes performed in the Garment Segmentation step. The left branch of the diagram corresponds to operations performed over the Saturation (S) channel, whereas the right branch of the diagram corresponds to operations over the Value (V) channel. The Hue (H) channel of the HSV image is not used. The results of the two threshold processes are merged together with a bitwise AND operation before applying the morphological operations.*

Finally, some morphological transformations are applied to the resulting mask to reduce noise due to false positives/negatives. A 5x5 square kernel is used in several closing operations, followed by a similar number of opening operations. Morphological transformations are operations applied to a binary image using a kernel or structuring element. The binary image is probed using the structuring element and pixels are written black or white depending on how this structuring element fits or misses the shapes in the image. Erosion and dilation are basic morphological transformations that reduce and increase the white elements present in the image according to how well the structuring element fits them. Closing and opening operations are based on the composition of these basic operations. A closing operation performs a dilation operation first, and then an erosion, removing holes present in the binary image. A opening operation, on the other hand, performs an erosion and then a dilation operation, removing small blobs and enlarging holes present in the binary image.

The output of this background subtraction step, a binary mask with background pixels represented as black and garment pixels represented as white, can be seen in Figure 4.3.

(a) *Original RGB image* **(b)** *Segmentation mask*

Figure 4.3: *Original RGB image captured by the robot sensor versus the binary segmentation mask obtained after the background subtraction process. Black pixels represent the background, whereas white pixels represent pixels belonging to the garment.*

4.2 Garment Approximated Polygon Computation

From the binary mask obtained in the previous step (section 4.1) a blob labeling algorithm is applied to detect the garment outline. This outline will be approximated to a simple polygon, and used in later stages to obtain the candidates to be a fold.

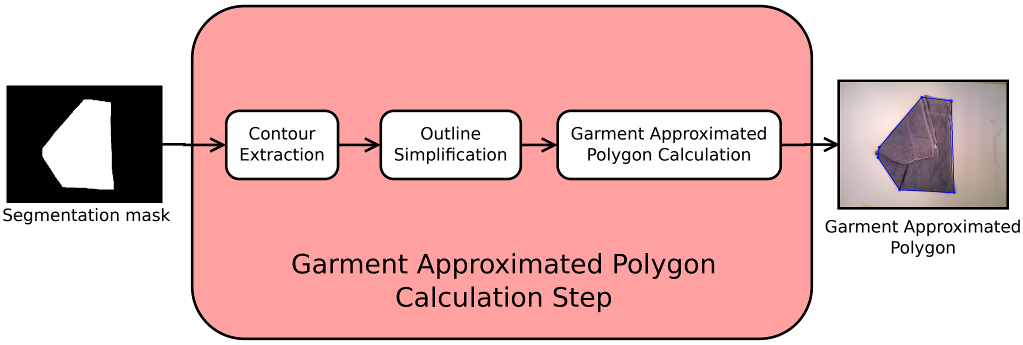


Figure 4.4: *Block diagram showing the different processes performed in the Garment Approximated Polygon Calculation step. The garment contour is obtained in the first process, and it is then simplified in the next processes to yield an Approximated Polygon that describes the garment.*

The contour extraction method used is the Topological Analysis by Border Following algorithm developed by Suzuki and Abe (Suzuki et al., 1985). It was selected as it is a widely used algorithm for connected-component labeling and countour finding. Other algorithms, such as Marching Cubes (Lorensen & Cline, 1987), work under the assumption that contours are isolines[1], which is not the case for binary masks, so they are not suitable. Only external contours are retrieved. A simple chain approximation is then applied to reduce the number of points contained in the contours, storing only the endpoints of the different segments that describe the garment outline.

Due to noise, sometimes some small blobs appear in segmentation masks, so the outline with the largest area is selected as the garment outline. This way,

[1]Curves along which a function has a constant value.

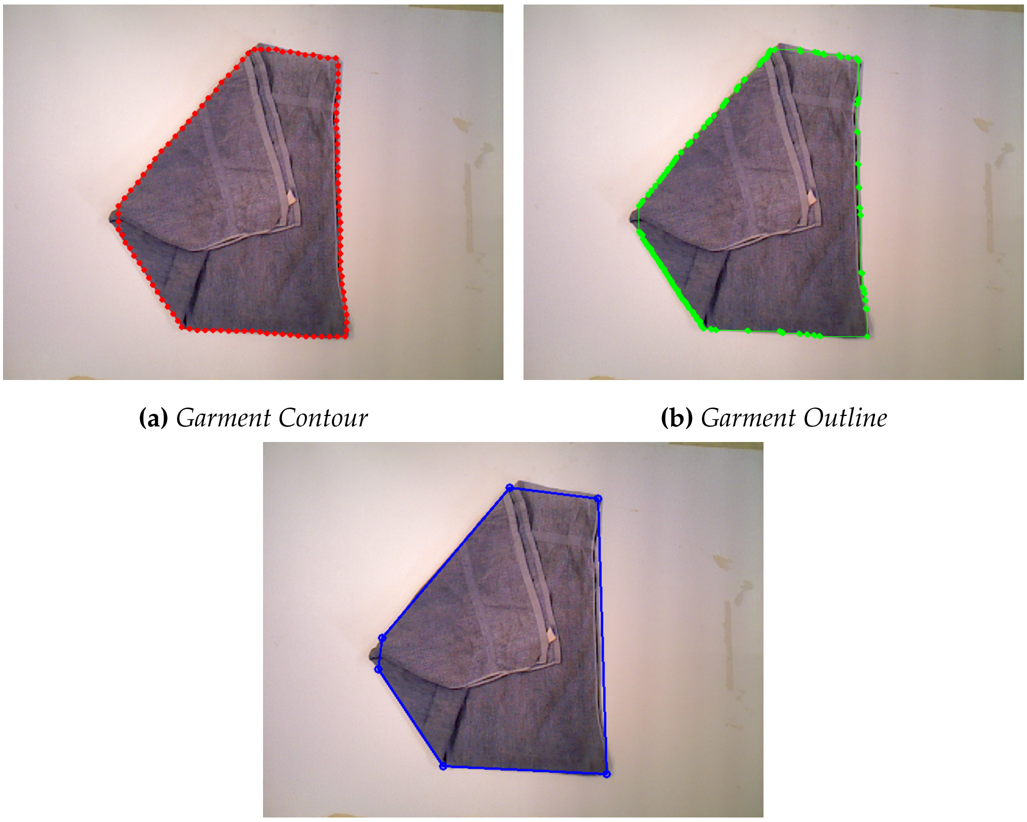
those small blobs are discarded.

After obtaining the garment outline, it is futher processed, as we want to obtain a simplified description of the garment outline. This simplified description is the garment approximated polygon. We assume the fold line has a very high probability of lying in the garment polygon and, therefore, this polygon will represent all the candidate segments to be a fold. To obtain the approximated polygon, the Ramer–Douglas–Peucker algorithm (Ramer, 1972; Douglas & Peucker, 1973) is applied. It is selected as it is efficient and a reliable implementation is available in the OpenCV[2] libraries used to implement this work. This algorithm recursively divides the outline in segments by choosing the first and last points of the curve and drawing a line. Then, it checks whether that point is closer to that line than a threshold $\epsilon > 0$ or not. If it is closer, all points not marked to be kept can be discarded; otherwise, if it is greater than ϵ, that point is marked to be kept and the procedure is repeated considering the last marked point as ending point. If there are no points left to process, the last point of the outline becomes the ending point again. The previous ending point becomes then the new starting point.

The parameter ϵ is calculated from the magnitude of the outline perimeter, considering it to be 1% of that value. The greater this value is, the more simplified the resulting polygon will be.

Figure 4.5 shows a comparison between the garment contour, the garment outline and the final approximated polygon.

[2]http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#approxpolydp, last accessed: March 22, 2016.



(a) *Garment Contour* **(b)** *Garment Outline*

(c) *Garment Approximated Polygon*

Figure 4.5: *Garment Contour, Garment Outline and Garment Appoximated Polygon. The Garment Contour contains all the points obtained from the segmentation mask. In the Garment Outline, points belonging to the same straight line are simplified and only the start and end points of the segment are used to represent it. After the Ramer-Douglas-Peucker algorithm is applied, a polygon with a low number of edges is obtained, which is used as Garment Approximated Polygon.*

Chapter 5

Garment Depth Map Clustering

This chapter explains in detail the Garment Depth Map Clustering stage that our algorithm performs to the garment data that has been previously segmented to group similar height regions of the garment. Some of these clusters represent overlapped regions of the garment. The block diagram of this stage can be seen in Figure 5.1.

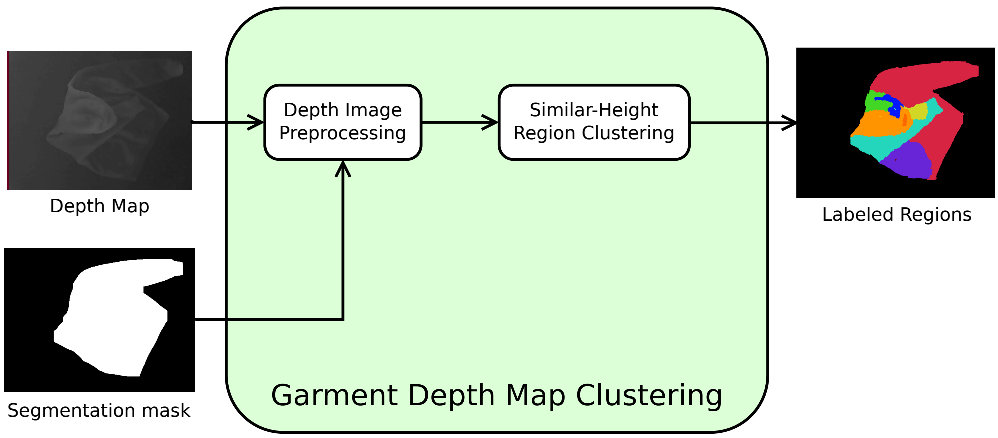


Figure 5.1: *Block diagram showing an overview of the Garment Depth Map Clustering stage and its different steps. The inputs of this stage are the depth map image and the segmentation mask obtained in the previous stage. These inputs are combined and preprocessed in order to apply the clustering step. The output of this stage are the labels of the different cluster regions.*

In this stage, the depth image data is preprocessed using the segmentation results from the previous stage to remove all depth data not related to the garment. Then, it applies a Watershed transform algorithm to find the different similar-height regions, that will be analyzed in the last stage to select the most suitable manipulation points to unfold the garment.

As mentioned before, some of these regions of similar height correspond to overlappìng regions of the garment, whose order can be determined using the average depth value of the pixels in each region. The garment region with the lowerst height value then corresponds to the region of the garment underneath those overlapping regions.

Once the different regions are labeled, the labeled clusters are passed to the next stage in order to determine the uppermost fold and the most suitable unfolding direction.

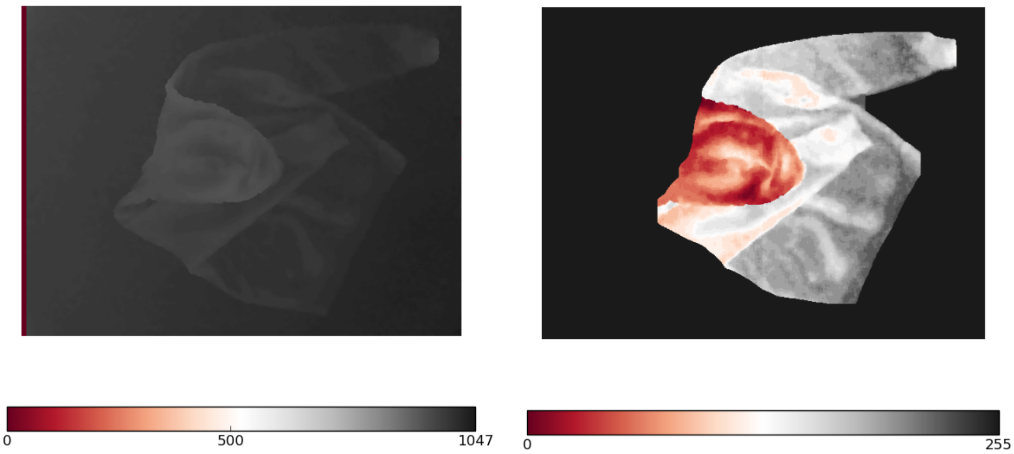
5.1 Depth Image Preprocessing

This step prepares the input depth data from the sensors to be used in later steps. The raw depth image data obtained from the robot RBG-D sensor cannot be used directly, as it contains data related to outliers and noise. The data coming from the table on which the garment lies is also present in this image. Therefore, this data requires some treatment before it can be processed in the next step (section 5.2).

For that purpose, the garment mask calculated in the previous stage, a required input of this step, is applied to the depth image of the garment. The objective is to discard the information related to the table. After this process only information about the garment is present on the depth image.

The depth input data is read expressed as a distance from the sensor to each pixel. As the next step will treat the depth data as a greyscale image, the remaining garment depth data needs to be normalized to the 8-bit range of a typical greyscale image. By normalizing the data we ensure that we are using the maximum resolution available (256 grey levels) to represent all the depth variability in the depth image. This helps the next step to find regions of similar height more accurately.

Figure 5.2 shows a comparison between the raw input depth data and the data after this processing step. The left image pixel values correspond to distances to the sensor expressed in mm, and the right image pixel values are those distances normalized to a 8-bit unsigned integer (256 values). The image on the left is shown mostly in grey shades, except from a thin red line seen on the left part of the image, corresponding to values out of sensor range. On the right image, representing the depth image after the preprocessing stage, it can be observed that the depth image now uses all the range available by the encoding as an 8-bit unsigned char.



(a) *Before preprocessing* **(b)** *After preprocessing*

Figure 5.2: *Depth image before and after preprocessing. Note that pixels closer to the sensor are shown in red, and pixels further from the sensor are displayed in grey. White values are intermediate values. The correspondence between colors and distance is explained in the colorbar beneath each figure.*

5.2 Similar-Height Region Clustering

Once the garment depth data is separated from the table depth data and normalized in the previous step, regions of similar height must be identified and labeled.

Grouping points according to a common feature, such as height, is a clustering problem, and several clustering methods and alternatives were described in section 3.2. A Watershed Transform (Digabel & Lantuéjoul, 1978) segmentation algorithm was selected to perform the clustering. This algorithm is faster than other superpixel approaches, and usually returns a single cluster per overlapping region, removing the need for a later merging step. This step is frequently required when working with superpixels to aggregate superpixels corresponding to similar regions, as they are conceived to divide the image in relatively small clusters of similar characteristics.

The Watershed Transform algorithm is a segmentation algorithm that considers a greyscale image as a topological surface where high intensity pixels correspond to peaks and hills, and low intensity pixels are equivalent to valleys. The algorithm fills the surface pouring water at each isolated valley. As the water level rises, the water from different sources will start to merge. To prevent them from merging, the algorithm constructs barriers at the merging regions, and continues this process of adding water and building barriers until all the peaks have been flooded. The resulting barriers are the segmentation regions border, and all the pixels within each region enclosed by these borders correspond to a different segmented item.

Figure 5.3 shows an example of applying the Watershed algorithm. The grey area represents a cross section of the image along a given image axis. Subfigure A shows the segmentation results when all local minima are selected as flooding sources (i.e. markers), and subfigure B shows the same results when one of those sources is omitted. The dotted lines on the right-hand side of the figure represent the catchment basins resulting from the Watershed Transform, and the colored regions are the different segments obtained.

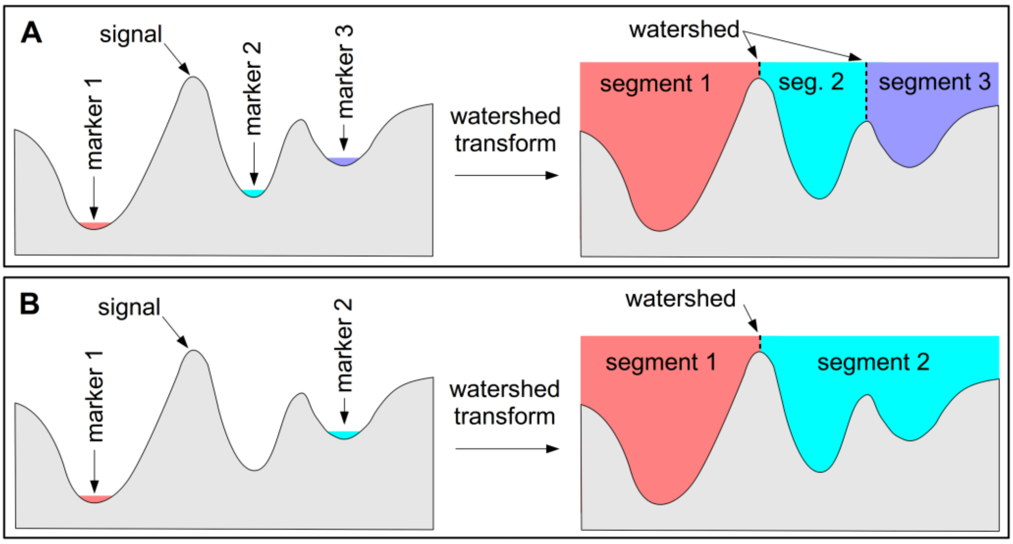


Figure 5.3: *Example of application of the Watershed Tranform algorithm. The grey area, marked as signal, represents the the pixel intensity along a given image direction, as if it was a cross section of the image. Figure extracted from (Fisher, 2014).*

A denoising process is performed before applying the Watershed transform, which is the actual clustering process. We use a Total Variation filter (Chambolle, 2004) to produce a smoother image while maintaining the edges sharp. The Total Variation filter works by minimizing the integral of the norm of the image gradient. As a result of this filter, piecewise-constant images ("cartoon-like" images) are obtained. This denoising process is required when using the Watershed transform algorithm, as it is an algorithm highly sensitive to local minima, which can be present due to noise or irregularities in the image.

The Watershed Transform segmentation algorithm is then applied to the depth image of the garment to locate the different parts that are overlapping each other. These regions are related to folded parts, that rest on top of other parts of the garment.

As said before, Watershed is very sensitive to local minima. Therefore, in practice, using local minima as sources for flooding leads to over-segmentation of regions. An enhanced version[1] of this algorithm allows the user to specify other criteria for selecting the seed points. The gradient of the greyscale depth-image is computed, and regions where the gradient has a low value are selected as these seed points. These regions correspond to homogeneous and continous regions, which are good candidates to be used as markers.

The different garment regions obtained with Watershed were labeled and used as input for the next step. Figure 5.4 shows the result of this process.

[1]http://scikit-image.org/docs/dev/auto_examples/segmentation/plot_marked_watershed.html, last accessed: March 22, 2016.

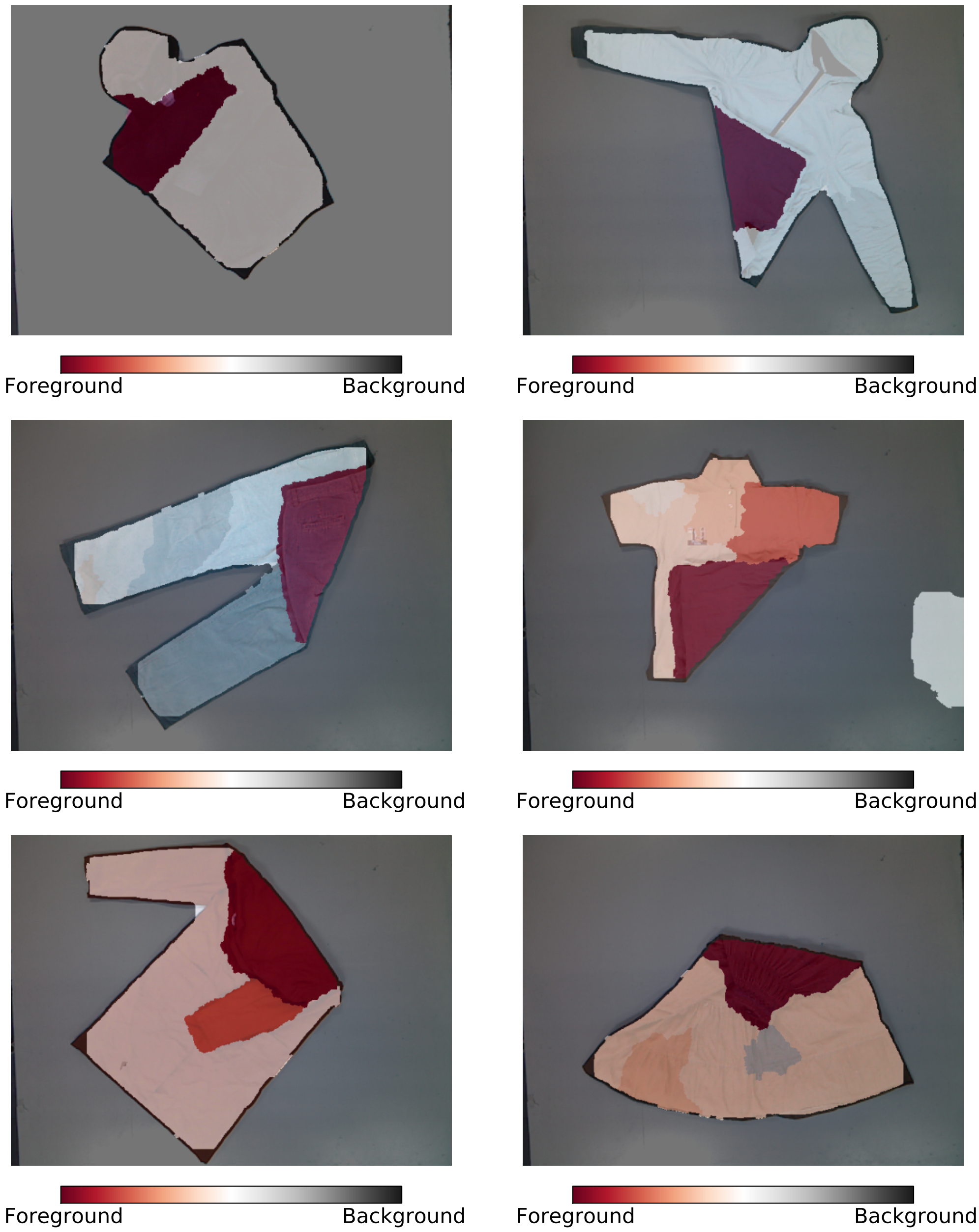


Figure 5.4: *The normalized depth images for different clothes are shown on the left side. The grey level is related to the height of the point as detected by the RGB-D sensor. A darker grey level indicates the points are closer to the sensor. On the right side, the labeled image returned by watershed algorithm is presented, where each color represents a region of similar height.*

Chapter 6

Garment Pick and Place Points

This chapter explains the last stage of the algorithm, in which the pick and place points required for unfolding the current fold are determined. These points will be later used by the humanoid robot to perform the unfolding operation. This stage has as input the clustered depth map obtained in the clustering stage (described in chapter 5), and the garment approximated polygon (calculated in section 4.2). Figure 6.1 shows the block diagram of the different steps that are performed in this stage.

6.1 Candidate Unfold Paths

Based on the assumption that when a garment has an overlapping fold, the fold line will rest on the garment contour, and the folded surface will have lower depth values (i.e. closer to the depth sensor), the next step in our algorithm is to create a set of paths from the highest point of the garment to the midpoint of each contour segment. These paths will be later analyzed using the clusters previously found in the depth image to select the path with less height variation.

To find the highest point in the garment, the clusters previously found with the Watershed algorithm are averaged using the median value for each cluster.

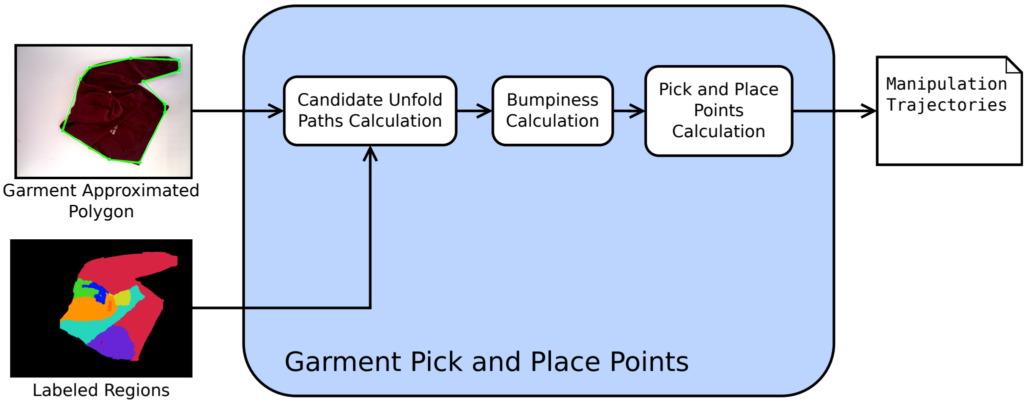
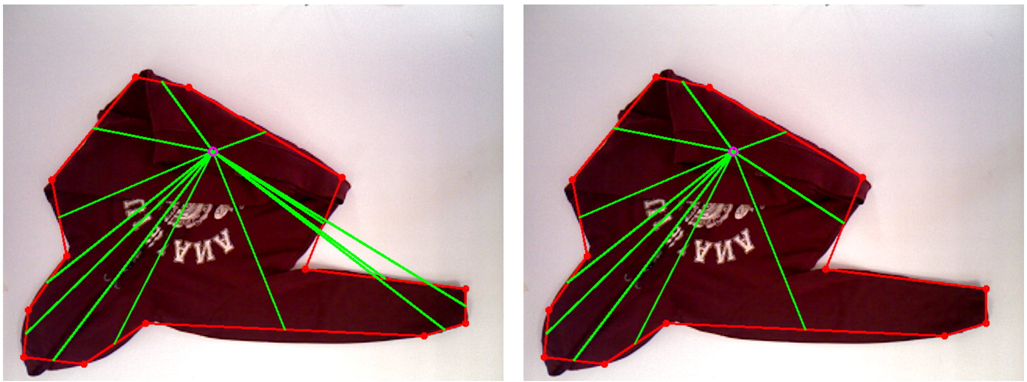


Figure 6.1: *Block diagram showing an overview of the Garment Pick and Place Points stage and its different steps. The inputs for this stage are the garment approximated polygon, obtained in the Garment Segmentation stage, and the labeled cluster regions, obtained in the Garment Depth Map Clustering stage. After the analysis of these inputs, the most suitable manipulation trajectories are output to the robot to perform the unfolding action.*

Then, the region with the lowest depth value, which is closest to the camera and therefore highest in the garment, is selected as overlapping fold based on the previous assumption. The centroid of the selected cluster is the point selected as the highest point. Using this method instead of selecting directly the highest point from the depth image increases the robustness of the algorithm against outliers and noise present in the depth image.

The midpoint of each segment of the garment approximated polygon is then computed, and a set of paths departing from the highest point and arriving to the midpoints is created.

These paths are checked so that they are located entirely inside the garment. Paths that go outside the garment approximated polygon are considered invalid. Figure 6.2 shows both the initial paths set and the unfold candidate paths set, without the invalid paths.



(a) *Set of candidate paths* **(b)** *Set of valid paths*

Figure 6.2: *The left image displays the set of initial candidate paths (green lines) to be used as unfolding directions. These paths are generated starting at the centroid of the highest region, represented with a magenta circle, and ending at the midpoint of each of the edges of the approximated garment polygon, shown in red lines. These paths are filtered, and the resulting valid paths, that do not cross the outer approximated polygon, are shown in the right figure.*

6.2 Bumpiness

Each of the candidate paths computed using the method explained in the previous section has to be analyzed in this step to determine the best unfolding path.

We can represent each of the n candidate paths obtained as a 2D parametric line ($\mathbb{R} \to \mathbb{R}^2$) depending on a parameter r, the radial distance to the highest point:

$$\text{path}(r) = [u(r), v(r)] \tag{6.1}$$

Where u and v are pixel coordinates in the image frame of reference.

Prior to the analysis, each of the paths of length L is discretized in segments with a constant length l. The depth image $\mathcal{D}(u, v)$ is sampled at those discrete points.

This results in n ordered sets $S = \{s_1, ..., s_m\}$ of depth samples to be analyzed, where:

$$s_i = \mathcal{D}(path(i \cdot l)), \quad i = 0, 1, 2, ..., m \tag{6.2}$$

As the different paths may differ in length L, the amount of sampled points ($m = \lfloor \frac{L}{l} \rfloor$) will be different for each path. Figure 6.3 shows n sample sets S_j corresponding to all of the candidate paths for a certain garment example.

The differences in depth between the overlapping fold region and the rest of the clothing article are assumed to be greater than the diferences in depth within the fold region points. Under that assumption, the metric to evaluate the best path is a *bumpiness* value B, which is calculated by penalizing the changes in depth along each of the n candidate sample sets S, as shown in Equation 6.3.

$$B = \sum_{i=2}^{m} |s_i - s_{i-1}| \tag{6.3}$$

The candidate set with the lowest bumpiness value, which corresponds to the path with the least and smallest height changes, is selected as the unfold direction. Figure 6.3 shows the candidate paths with their corresponding height profiles, calculated over a blanket in early stages of the development of the algorithm.

As an example of the computation of this *bumpiness* value, let us imagine a simple garment with a single overlapping fold. The Garment Clustering Stage may reveal two clusters: the fold overlapping the garment, with a median height value of 2 cm, and the rest of the garment beneath the fold, with a median height value of 1 cm. In the depth map, the overlapping cluster would be surrounded by pixels corresponding to the garment region in all its edges but the one corresponding with the folding edge. After the formula is applied to all candidate paths, a *bumpiness* value of $B = |2 - 1| = 1$ is obtained in all paths except the one arriving to the folding line, whose *bumpiness* value is $B = 0$, as no height changes occur. The path with minimum B ($B = 0$) is the one selected as unfolding direction.

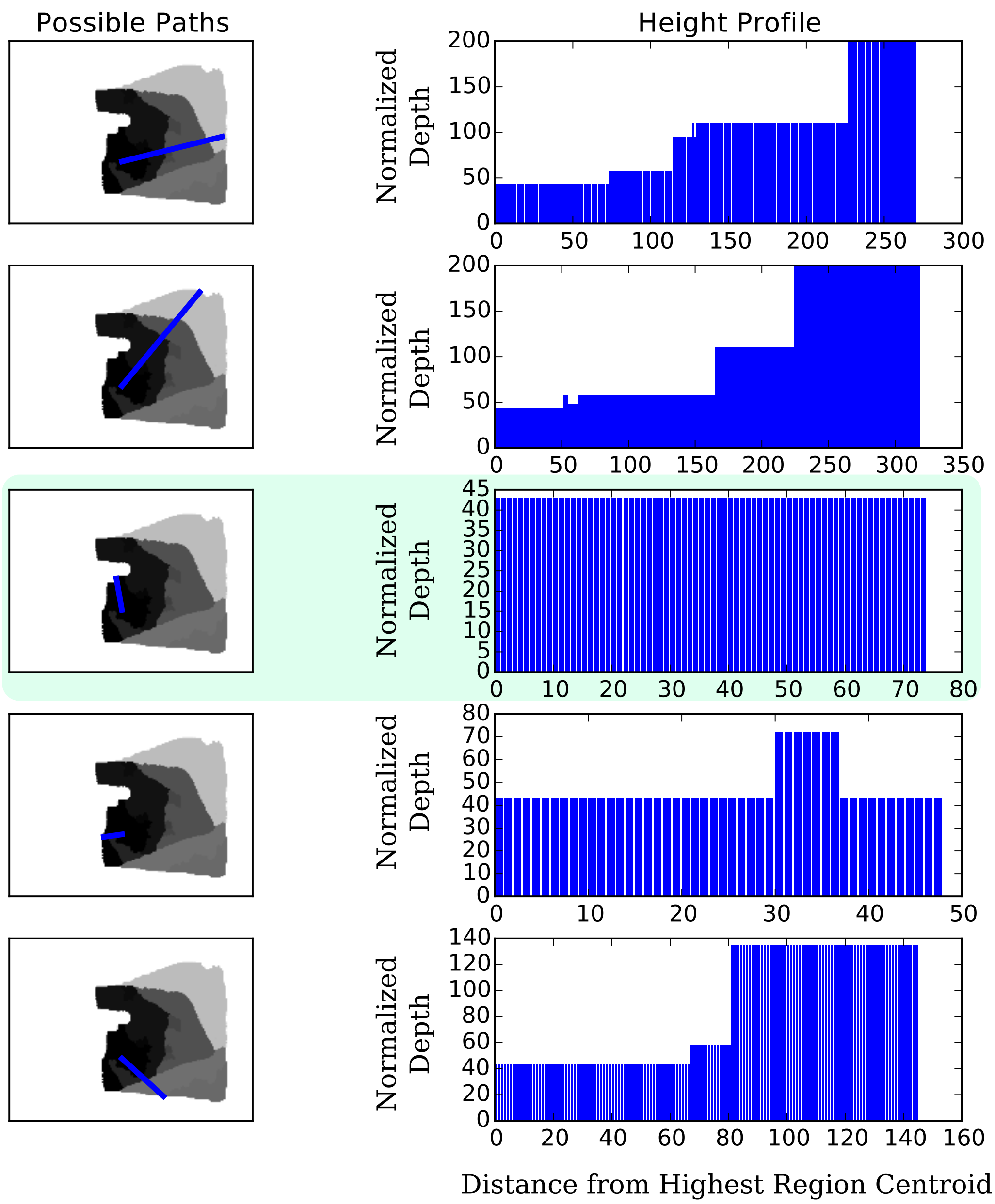


Figure 6.3: *On the left side, the candidate paths are shown. On the right side, the height profile of each path is shown. Note that the depth sensor computes the distance to the object from itself, so that a low value in the bar plot means a closer object to the sensor, and therefore it is a region with more height with respect to the table. The highlighted path is the one with the lowest value of bumpiness, and it is selected as suitable unfolding path. These values were calculated over a blanket in early stages of the development of the algorithm.*

6.3 Pick and Place Points

Once the most promising path for unfolding has been identified, the last step is to obtain pick and place points for the robot to perform the unfold action. Due to the absence of a cuantitative metric for determining the suitability of those points, the point selection criteria has been selected using intuition from all possible choices.

The selected grasping point for the picking action is located at the intersection between the unfolding path direction line and the highest garment region border. This point is obtained by calculating the intersection between the selected path line and the highest region contour. The operation results in two intersection points, from which the furthest to the garment border is selected.

The other intersection point is used as axis for a point reflection. To find the placing point, a reflection transformation is applied to the picking point using the aforementioned point as axis. Figure 6.4 shows the unfold directions for several clothes, departing at the picking points and arriving to the placing points.

Figure 6.4: *Final directions calculated for each garment provided to the system. Each arrow departs where the robot should pick the fold and arrives where it should be placed.*

Chapter 7

Experiments and results

The algorithm presented in this work was implemented and evaluated through several experiments using our own garment dataset, as well as validated with experiments using a real robot. This chapter is devoted to explaining the experiments performed using the algorithm presented in this thesis. It includes the implementation details, the evaluation dataset and experiments details and the validation experiments description. It also includes the results drawn from the different experiments performed.

7.1 Implementation details

The software implementation for this thesis is divided into three different parts dedicated to data obtention, data processing (the algorithm itself) and results evaluation.

The first part is in charge of communication with the sensor with the objective of extracting the depth data. This communication is performed using YARP (Metta, Fitzpatrick, & Natale, 2006). YARP is a set of libraries, protocols, and tools that leverages many common tasks required for a humanoid robot to work. The tasks include actuator control and command, communications with the robot and between software parts, and accessing to data captured from different common sensors, such as depth sensors.

The second main part is the implementation of the unfolding algorithm. This implementation was executed using Python[1] as our language of choice. Python is a programming language that allows a quick development of prototype software, and has a large collection of external modules to perform several tasks, from math calculations to computer vision. The implementation is divided into 3 consecutive stages, and within each of the stages, each step is also isolated and executed consecutively, to avoid tight coupling between the different stages and steps of the algorithm. This will allow the authors to quickly explore and test alternative implementations for each of the stages, or even for a single step within any stage, and check whether they improve the performance of the algorithm or they do not cause any significant improvement. The current implementation is based on two different computer vision libraries: OpenCV[2] and Scikit-image[3]. This is due to the fact that during the development of this work several algorithms were tested, and each library contains only a subset of those algorithms. In the latest version, OpenCV is used for basic computer vision and Scikit-image for more advanced algorithms such as Watershed[4] and other superpixel-based clustering methods. The implementation also relies heavily on Numpy[5]. Numpy is the *de facto* Python library for scientific computing, and all the operations with trajectories and unfolding paths required in the work, such as line intersections or line perpendicularity, were implemented using the mathematical functions present on Numpy. The unfolding algorithm implementation has been released to the public with an Open Source license, and it is already available online[6].

[1] http://www.python.org, last accessed: March 22, 2016.
[2] http://www.opencv.org, last accessed: March 22, 2016.
[3] http://scikit-image.org, last accessed: March 22, 2016.
[4] http://scikit-image.org/docs/dev/auto_examples/plot_watershed.html, last accessed: March 22, 2016.
[5] http://www.numpy.org/, last accessed: March 22, 2016.
[6] https://github.com/roboticslab-uc3m/textiles, last accessed: March 22, 2016.

For the evaluation of the results, a Graphical User Interface (GUI) was created using Python and PySide[7] (LGPL Qt bindings for Python). As it is shown in figure 7.1, this GUI is composed of a main view where results are presented and 3 buttons to indicate the 3 possible evaluations that each result can be given (Bad, Fair and Good). These evaluations will be explained in further detail later in this chapter, in section 7.2.

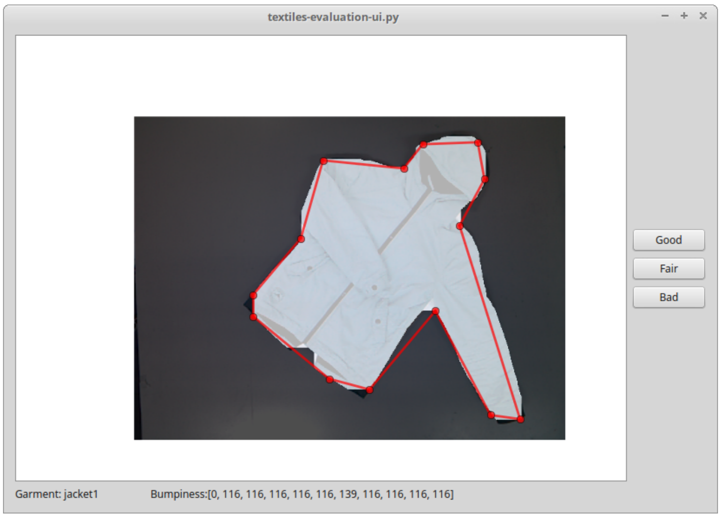


Figure 7.1: *GUI developed in PySide to semi-automate the evaluation of the algorithm*

Some extra information is also provided in the bottom part of the GUI, such as the name of the garment being evaluated and the bumpiness value for all the candidate unfold paths. Results from each of the 3 stages of the algorithm are first calculated by running the algorithm with all the garment samples. Then, these results are presented consecutively for all the samples in the dataset, and if any of them is evaluated as Bad, the remaining ones are skipped for that garment, and automatically evaluated as Discarded. All these evaluations are logged and when there are no results left to score, the statistic analysis of the logged evaluations is performed and presented to the evaluator.

[7]https://wiki.qt.io/PySide, last accessed: March 22, 2016.

7.2 Algorithm Evaluation

Previously to testing on the real robot, the algorithm was evaluated using a garment dataset presenting folded garment samples from several different garment categories. These results are later scored with a qualitative metric and then analyzed. This section addresses the garment dataset, the qualitative metric developed, and the analysis of these results.

7.2.1 Garment Dataset

To test the computer vision algorithm, a garment dataset was required. A few garment datasets meant for garment manipulation with robots already exist, such as the garment datasets from the CloPeMa project, that have been all made publicly available[8] to foment algorithm benchmarking. These garment datasets are composed of different types of sensor information (e.g. stereo pairs, depth images, etc) from different kinds of garment samples. There are 4 main datasets, which include samples of spread garments, garments in different stages of a folding sequence, and garments being manipulated by a robot arm, both static and in movement. However, after inspection of the samples in these datasets the author concluded that they are not suitable to test our algorithm, since these datasets are mainly focused on modeling and folding garments once they are already extended. This work is focused on unfolding garments to arrive to the spreaded, unfolded state, so it requires the garments to be flat and present some overlapped regions (folds).

As none of the existing datasets fit the objectives of this work, a new folded garment dataset was crafted[9]. The focus of this new dataset is to provide samples of garments placed on a flat surface, and presenting some amount of folds. This pose simulates the result of randomly spreading a garment over a table after picking it from a pile and prior to folding the garment. To assure some ex-

[8]http://clopema.felk.cvut.cz/public_data.html, last accessed: March 22, 2016.
[9]http://tinyurl.com/garments-birdsEye-zip, last accessed: March 22, 2016.

tent of variability within the dataset, the samples were taken from garments belonging to 6 different and representative garment categories: skirt, jacket, pants, polo, robe and hoodie. There is a total of 120 samples of different folds in garments from the mentioned categories. Each category set is composed by 20 samples, 10 of which present one fold and 10 of which present two folds in the garment.

This data was obtained using an ASUS Xtion PRO LIVE with RGB and depth channels set at 640x480 resolution. The sensor was placed on top of the working surface providing a bird's eye view over the garment folding environment, with its image plane almost parallel to the working surface. The garments were placed over the surface and folded, and the RGB-D data was then captured for that pose. This procedure was repeated 20 times with different folds for each of the 120 garments that compose the dataset.

7.2.2 Experiments

Using the garment dataset described in the previous section, the algorithm was applied to each of the dataset samples, recording the output at each of the 3 stages of the algorithm. Figures 7.2, 7.3 and 7.4 show the final output of the algorithm and the computed unfolding directions for 4 garments of each of the 6 categories.

For evaluating the algorithm, a scoring metric has been established. The criteria for the scoring is qualitative, due to the lack of an objective ground truth to provide quantitative comparisons. The output of each stage is manually classified into one of the following categories: Bad, Fair, Good and Discarded. Results from one stage that prevent the next one to work properly are classified as Bad, and the next stages for that item are classified as Discarded and skipped. Results that are not perfect, but allow the next stage to compute its output are classified as Fair. Note that the Fair score in the last stage means that the robot could unfold it, but it is not the optimal path. Finally, perfect results for a stage are given

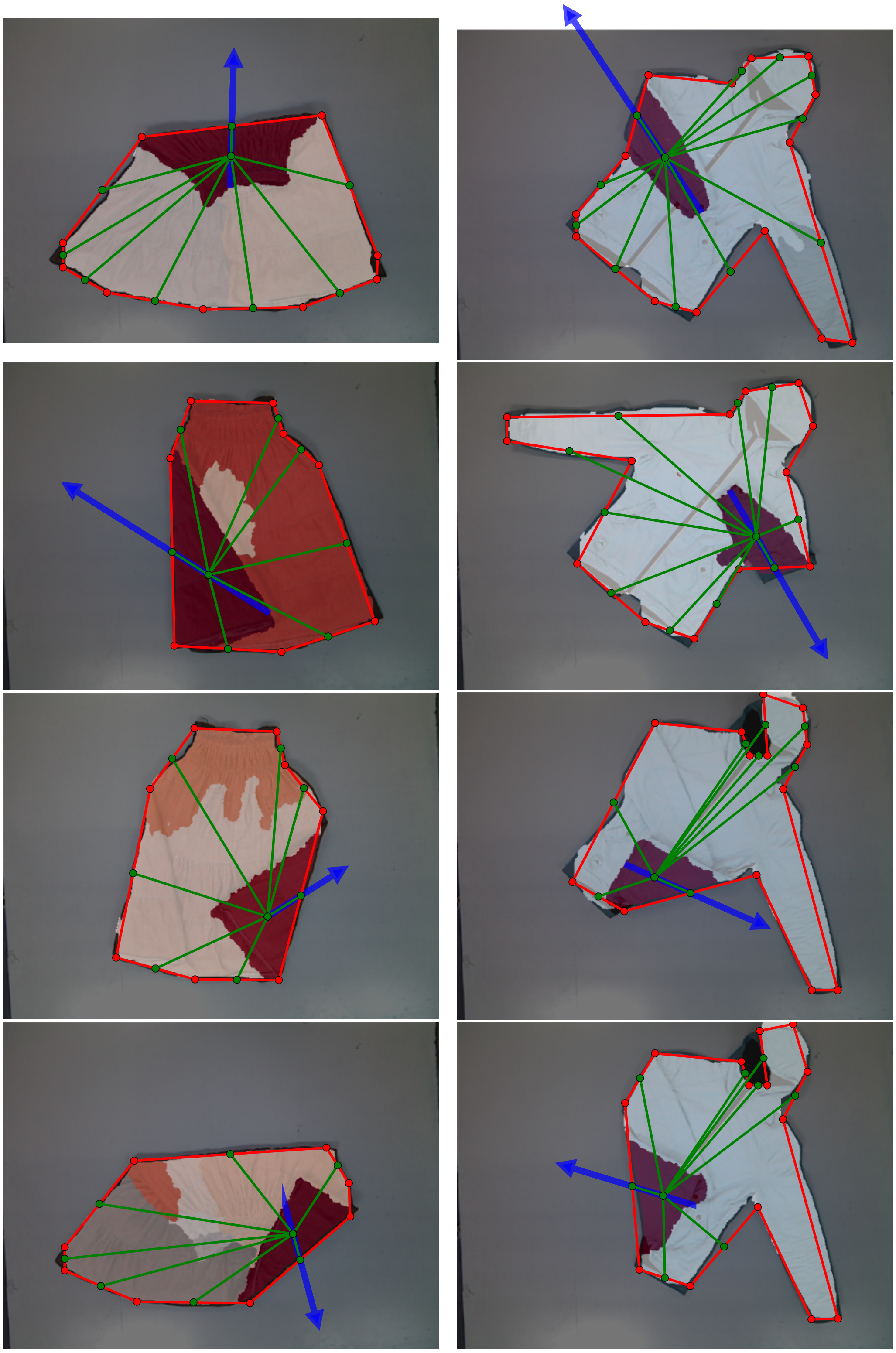


Figure 7.2: *Final output of the algorithm and the computed unfolding directions. Each column includes the output corresponding to 4 of the 20 database samples for each of the 6 garment categories considered. This figure includes the categories Skirt and Jacket.*

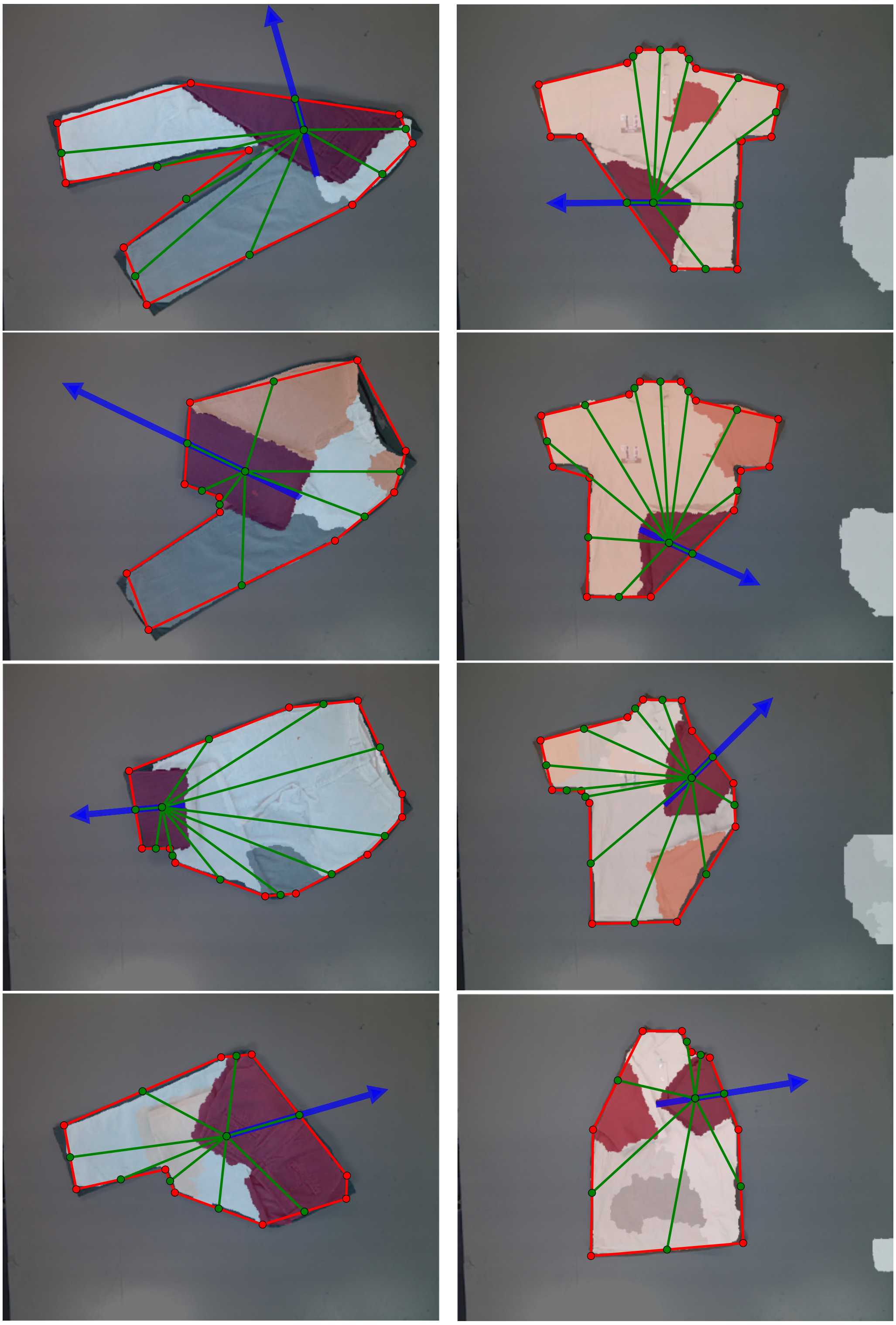


Figure 7.3: *Final output of the algorithm and the computed unfolding directions. Each column includes the output corresponding to 4 of the 20 database samples for each of the 6 garment categories considered.This figure includes the categories Pants and Polo.*

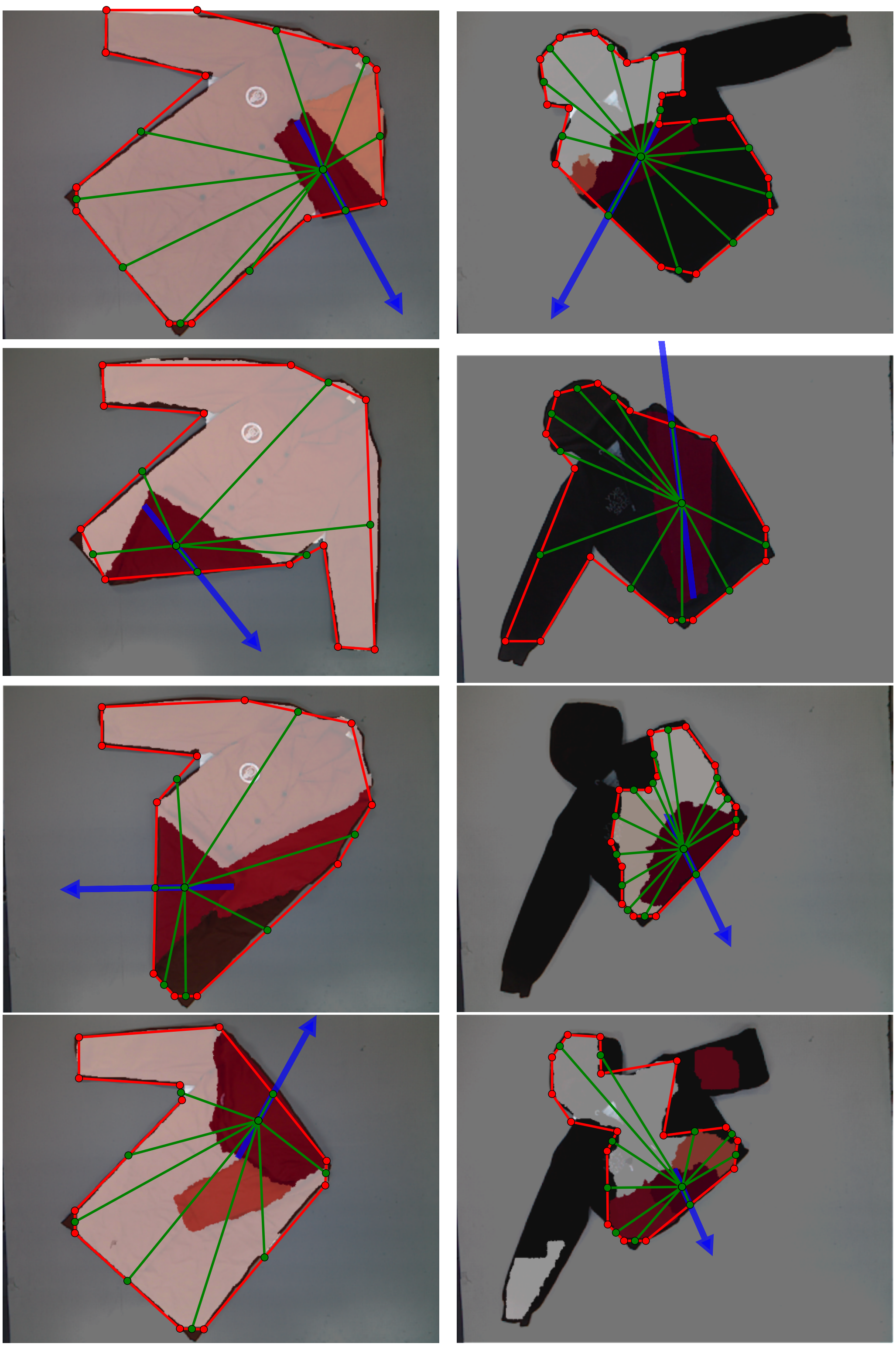


Figure 7.4: *Final output of the algorithm and the computed unfolding directions. Each column includes the output corresponding to 4 of the 20 database samples for each of the 6 garment categories considered. This figure includes the categories Robe and Hoodie.*

the Good score.

Table 7.1 shows the raw data of this scoring, indicating the amount of samples that received each score at a given stage for each category.

	Stage	Bad	Fair	Good	Discarded
Skirt	Segmentation	0	0	20	0
	Clustering	3	6	11	0
	Pick & Place	5	5	7	3
Jacket	Segmentation	0	9	11	0
	Clustering	8	8	4	0
	Pick & Place	3	3	6	8
Pants	Segmentation	1	1	18	0
	Clustering	7	5	7	1
	Pick & Place	8	1	3	8
Polo	Segmentation	0	17	3	0
	Clustering	3	13	4	0
	Pick & Place	1	4	12	3
Robe	Segmentation	0	0	20	0
	Clustering	5	7	8	0
	Pick & Place	3	3	8	6
Hoodie	Segmentation	11	3	6	0
	Clustering	6	2	1	11
	Pick & Place	3	0	0	17

Table 7.1: *Raw data results of the established scoring metric for each garment category and algorithm stage*

Results in each row must add up to 20, since all the garments from each category in the dataset are to be evaluated in each stage. Note also that Bad results from previous stages are added to the amount of Discarded samples in the last column. The Discarded samples have not been presented to the user for evaluation, since previous Bad results prevent next stages to yield sucessful results.

A stage by stage analysis is provided in Table 7.2. Each cell represents the percent of Fair and Good qualified samples with respect to the amount of samples that actually reached the given stage:

$$cell = \frac{Fair + Good}{Bad + Fair + Good} \cdot 100\% \tag{7.1}$$

The Overall performance included in the last row shows the percent of Fair and Good samples taking into account all the samples, including the ones scored as Discarded, and corresponds to the product of the percentages of each individual stage, since they are independent events.

Stage / Category	Skirt	Jacket	Pants	Polo	Robe	Hoodie	All
Segmentation	100	100	95	100	100	45	**90**
Clustering	85	60	63.2	85	75	33.3	**70.4**
Pick & Place Points	70.6	75	33.3	94.1	78.6	0	**69.3**
Overall	60	45	20	80	55	0	**43.3**

Table 7.2: *Results analysis per stage and garment category, expressed in percentage (%)*

For instance, the Segmentation stage was evaluated as having performed correctly (which includes Fair and Good samples) for 100% of the Skirt category. With the remaining samples, the Clustering stage performance was evaluated. In this case, within the Skirt category, all of the Skirt samples were evaluated, because they had all passed the previous stage. In this evaluation, 85% achieved a passing score (Fair and Good). This set of samples was passed on to the Pick & Place points stage, and for these samples, 70.6% achieved a passing score within the Skirt category. The Overall algorithm performance for the category is computed as the product of the percentages of three stages: $1 \cdot 0.85 \cdot 0.70 = 60\%$

based on the 20 skirt samples. Table 7.2 includes an additional column which applies this process based on the evaluation of All the 120 samples of the dataset.

Analyzing the results seen in Table 7.2, we can conclude that the Segmentation stage of the algorithm has provided successful results. This stage only achieved low evaluations for the Hoodie, which is black. This might be due to the fact that as this stage relies in color to perform the garment segmentation. As it is using the saturation channel of the HSV color space to detect colorful pixels, and black pixels can have any saturation value, some pixels may be considered not belonging to the garment. The Clustering stage of the algorithm provides a 70.4% success rate on the evaluated samples. Certain cluster contours are not fully contained within the region they represent. This may be due to filters used between stages for noise reduction, that may produce a blurring side-effect, and may be adjusted. A fine tuning of Watershed parameters to alter the cluster size could also improve these results. Finally, the Pick and Place Points stage results show our algorithm finds suitable unfolding directions for 69.3% of the evaluated samples. Regarding the current pick point, the highest garment point or highest region centroid could be used and tested as alternatives. Additionally, instead of using the fold midpoint as a symmetry point to compute the current place point, the entire fold could be used as a symmetry axis for this computation.

7.3 Algorithm Validation

Once the algorithm has been evaluated successfully on the garment dataset, experiments with a real robot were performed to validate the unfolding algorithm. Some clothing articles were provided to the robot over a flat surface, and the robot applied the unfolding algorithm to the input data from the depth sensor to compute the most suitable pick and place points, and then perform an unfolding operation. This section will describe in further detail the experimental setup and the experiments that were executed with the robot.

7.3.1 Experimental Setup

The experimental setup used to perform the evaluation tests consisted of several elements in a laboratory environment: a flat surface, a depth sensor and a humanoid robot. Figure 7.5 depicts this setup.

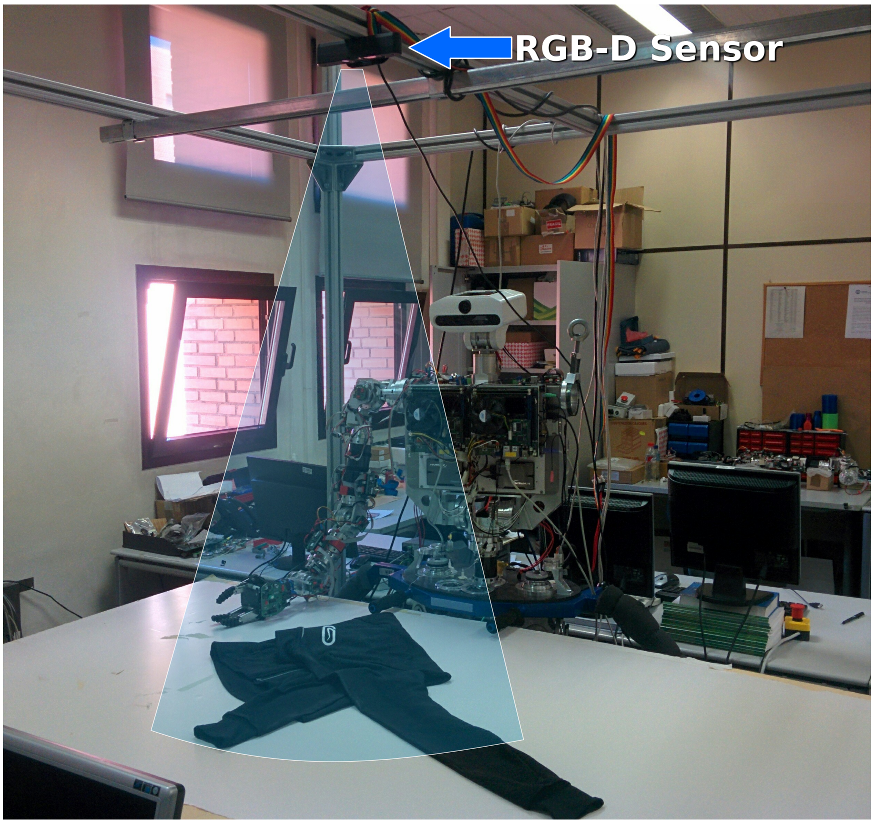


Figure 7.5: *This figure shows the experimental setup used to test the algorithm. The garment rests over a flat white table, with the RGB-D sensor on top. Besides the table, the humanoid robot TEO waits for manipulation trajectories to indicate it how to perform the unfold action.*

The garment was placed on a white, flat surface, parallel to the floor. Over the garment, an ASUS Xtion PRO LIVE depth sensor was attached to a structure to capture data from a top view, in a position similar to the one used to generate the garment dataset.

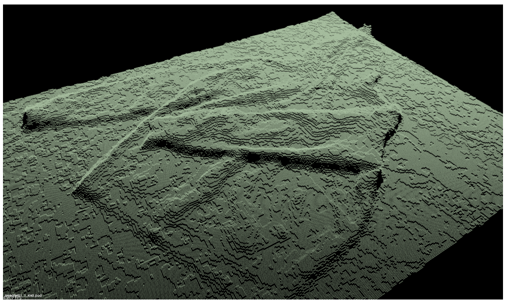
This final set of laboratory experiments to validate the unfolding algorithm were performed using the full-size humanoid robot TEO (Martínez et al., 2012). The robot's gripper, designed with passive compliance and relatively large objects in mind (e.g. fruit, bottles, etc), was substituted by a gripper for garment manipulation actuated with hobby servomotors. This gripper was 3D printed in PLA plastic, and controlled with an Arduino board connected by USB to the robot's manipulation computer.

7.3.2 Experiments

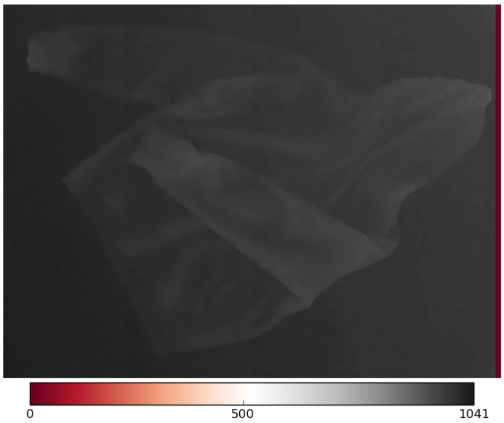
The set of experiments consists in providing several folded garments to the humanoid robot which it has to analyze using the unfolding algorithm presented in this thesis. After the most suitable pick and place points are computed, the humanoid robot should perform and unfolding operation.

The starting point of each experiment is the data adquisition process. The garment data was captured using the ASUS RGB-D sensor in the same configuration (at the ceiling) as for the dataset generation. As garment data is obtained as a point cloud, it needs to be converted to a depth map image for its later analysis. This conversion is done by simply using the z component of each point as the depth value for each pixel of the depth image. If there were a known deviation from the surface (affecting perpendicularity), the depth image could be recovered from the point cloud using the instrinsic and extrinsic parameters of the sensor instead. As the sensor is placed on top of the garment, perpendicular to the surface on which the clothing article rests, both methods yield very similar results, as shown in figure 7.6. The RGB values are also recorded for each depth image, obtaining an RGB-D image.

Then, the 3 stages of the unfolding algorithm presented in this work are applied consecutively, obtaining the most suitable pick and place points to manipulate the garment. Once the unfolding pick and place points are computed, depth sensor coordinates are converted to the robot root frame. A standard pick



(a) *3D Point cloud*



(b) *Depth Map*

Figure 7.6: *Comparison of the 3D point cloud with the Depth map obtained from its projection onto the table. The colorbar beneath the second figure shows how depth from the camera (in mm) is mapped to the figure colors.*

and place operation is performed by the robot with the final obtained points.

Figure 7.7 shows the robot performing the unfolding operation.

Figure 7.7: *Garment unfolding operation being performed by the Teo humanoid robot. Frames are time-ordered from left to right and from top to bottom.*

Chapter 8

Conclusions and future work

This chapter will describe the main contributions made to the state of the art by this work. Along the development of this work and during the experiments some challenges appeared. They will be covered after the main contributions along with future work lines that address those challenges and further improve the state of the art.

8.1 Main contributions

This section is dedicated to explain the main contributions of this work, and develop them in detail according to the different steps in the unfolding algorithm.

The main contribution of our work is a model-free garment-agnostic algorithm that can compute the pick and place points for a manipulator robot to iteratively unfold a garment, so that other algorithms can then determine the garment category and apply the folding sequence. This algorithm presents the following advantages:

- It provides a general method for detecting folds in deformable objects without a prior model of the garment to be unfolded.

- It computes the best position of the pick point, direction of movement, and place point in order to unfold the detected fold.

- As it uses depth data, it is independent of the color or patterns present in the garment, except for background extraction.

Listed by their corresponding stage, the main contributions of this work are:

1. **Garment Segmentation:** Regarding garment segmentation, he main contribution is a segmentation stage that is independent of the shape and color of the garment. This stage works as long as the table that holds the garment is white or grey and the garment is more colorful than the table, which is a reasonable assumption for the lab environment used to evaluate this work. The segmentation stage, additionally, requires no user input to label background/foreground samples, so the unfolding process can be applied autonomously, without any human operator.

2. **Garment Depth Map Clustering:** The algorithm improves existing approaches by using Watershed as clustering algoritm. Using Watershed allows our algorithm to obtain the different contiguous regions directly, without the merging step that it is required when using other superpixel-based clustering methods. For the same reason, it also removes the need for a threshold value for labelling contiguous superpixel regions. The absence of a threshold value avoids breaking similar height regions into different label regions due to large wrinkles, so that later unwrinkling stages are not required either.

3. **Garment Pick and Place Points:** The main contribution of this work regarding choosing pick and place points is that the selection of those points is independent of the garment category. Previous work found in the literature required the garment category or model to be specified or learnt to have a prior knowledge of the most suitable grasping points.

According to the evaluation of the algorithm with our garment dataset, suitable pick and place points were generated successfully for 43.3% of all the garments in the dataset, with a 90% of success in the Garment Segmentation stage, a 70.4% of sucess in the Garment Clustering stage and a 69.3% of success in the Garment Pick & Place Points stage.

8.2 Future Work

During the development and evaluation of this work some challenges arised, providing the author opportunities to further develop and improve this work. This section will introduce the challenges and the opportunities to find solutions to address them, regarding each of the stages that compose the algorithm.

1. **Garment Segmentation:** As stated in the chapter devoted to experiments and results, the Garment Segmentation stage performs correctly for most cases, with the exception of black garments. In the HSV color space being used the representation of black pixels is unstable for the saturation channel, as a black pixel can have any saturation value. This confuses the current segmentation algorithm, based on the saturation channel. Future works can address this problem in several ways, for example, with more complex rules for thresholding the HSV channels. Using a segmentation stage based only in depth data, therefore eliminating all need for color information is another potential approach, that would also increase the robustness of the Garment Segmentation stage against changes in light conditions.

2. **Garment Depth Map Clustering:** The results obtained with the garment dataset indicate that in some cases the regions found with the Watershed Transform algorithm do not exactly overlap with the underlying garment regions. Fine-tuning of the Watershed parameters, as well as the filters applied on this stage are likely to increase the accuracy of the labeled over-

lapping regions. Moreover, in general we have found that working with a single point of view is a very limiting factor, as occlusions sometimes make folds ambiguous. As the presented approach does not depend on garment models, dissambiguating these situations is very challenging. For that reason, the author strongly believes that moving to an approach that uses a 3D point cloud of the garment as input data would have more information available to solve those ambiguous situations in a better way. Being able to use 3D information would also remove the current limitation of this approach to folds with the fold edge coincident with the surrounding garment approximated polygon.

3. **Garment Pick and Place Points:** While the selected points may serve as a rule-of-thumb for robotic system developers, there is no clear metric for cuantitatively evaluating the suitability of the selected points. Therefore, experimental validation through further experiments with robotic systems is required to determine whether a better pick and place strategy exists:

   - For instance, the highest garment point or highest region centroid could be used as alternative to the current pick point, under the assumption that that point would correspond to an overlapped garment region not attached to the garment regions underneath.
   - Other place points could also be chosen depending on a different criteria. For example, the place point could be calculated using the fold line as axis of symmetry, or determined based on the unfolding trajectory selected.
   - The trajectory used to unfold the garment is a simple one, which follows a straight line connecting the pick point, a point above the pick point, a point above the place point and the place point. More elaborated trajectories, such as splines or curves, could be used instead,

such the one Li et al. present in their method for folding deformable objects (Li, Yue, Xu, Grinspun, & Allen, 2015).

Finally, as the validation experiments were performed using a limited set of garments in our laboratory, we would like to perform extensive testing of the algorithm on a wider set of garments with both our full-body humanoid robot and more traditional industrial manipulators, to compensate for the lack of disponibility of humanoid robots in current industrial environments.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2010). *Slic superpixels* (Tech. Rep.).

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *34*(11), 2274–2282.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, *20*(1-2), 89–97.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(5), 603–619.

Cusumano-Towner, M., Singh, A., Miller, S., O'Brien, J. F., & Abbeel, P. (2011). Bringing clothing into desired configurations with limited perception. In *Robotics and automation (icra), 2011 ieee international conference on* (pp. 3893–3900).

Digabel, H., & Lantuéjoul, C. (1978). Iterative algorithms. In *Proc. 2nd european symp. quantitative analysis of microstructures in material science, biology and medicine* (Vol. 19, p. 8).

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *10*(2), 112–122.

Doumanoglou, A., Kargakos, A., Kim, T.-K., & Malassiotis, S. (2014). Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In *Robotics and automation (icra), 2014 ieee international conference on* (pp. 987–993).

Doumanoglou, A., Kim, T.-K., Zhao, X., & Malassiotis, S. (2014). Active random forests: An application to autonomous unfolding of clothes. In *Computer vision–eccv 2014* (pp. 644–658). Springer.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).

Fisher, A. (2014). Cloud and cloud-shadow detection in spot5 hrg imagery with automated morphological feature extraction. *Remote Sensing*, *6*(1), 776.

Kita, Y., Saito, F., & Kita, N. (2004). A deformable model driven visual method for handling clothes. In *Robotics and automation, 2004. proceedings. icra'04. 2004 ieee international conference on* (Vol. 4, pp. 3889–3895).

Kita, Y., Ueshiba, T., Neo, E. S., & Kita, N. (2009). Clothes state recognition using 3d observed data. In *Robotics and automation, 2009. icra'09. ieee international conference on* (pp. 1220–1225).

Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., & Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *31*(12), 2290–2297.

Li, Y., Chen, C.-F., & Allen, P. K. (2014). Recognition of deformable object category and pose. In *Robotics and automation (icra), 2014 ieee international conference on* (pp. 5558–5564).

Li, Y., Wang, Y., Case, M., Chang, S.-F., & Allen, P. K. (2014). Real-time pose estimation of deformable objects using a volumetric approach. In *Intelligent robots and systems (iros 2014), 2014 ieee/rsj international conference on* (pp. 1046–1052).

Li, Y., Xu, D., Yue, Y., Wang, Y., Chang, S.-F., Grinspun, E., & Allen, P. K. (2015). Regrasping and unfolding of garments using predictive thin shell modeling. In *Proceedings of the ieee international conference on robotics and automation (icra).*

Li, Y., Yue, Y., Xu, D., Grinspun, E., & Allen, P. K. (2015). Folding deformable objects using predictive simulation and trajectory optimization. In *Proceedings of the 27th ieee/rsj international conference on intelligent robots and systems (iros).*

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Acm siggraph computer graphics* (Vol. 21, pp. 163–169).

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297).

Mariolis, I., & Malassiotis, S. (2013). Matching folded garments to unfolded templates using robust shape analysis techniques. In *Computer analysis of images and patterns* (pp. 193–200).

Mariolis, I., & Malassiotis, S. (2015). Modelling folded garments by fitting foldable templates. *Machine Vision and Applications*, *26*(4), 549–560.

Martínez, S., Monje, C. A., Jardón, A., Pierro, P., Balaguer, C., & Muñoz, D. (2012). Teo: Full-size humanoid robot design powered by a fuel cell system. *Cybernetics and Systems*, *43*(3), 163–180.

Metta, G., Fitzpatrick, P., & Natale, L. (2006). Yarp: yet another robot platform. *International Journal on Advanced Robotics Systems*, *3*(1), 43–48.

Miller, S., Fritz, M., Darrell, T., & Abbeel, P. (2011). Parametrized shape models for clothing. In *Robotics and automation (icra), 2011 ieee international conference on* (pp. 4861–4868).

Mori, G. (2005). Guiding model search using segmentation. In *Computer vision, 2005. iccv 2005. tenth ieee international conference on* (Vol. 2, pp. 1417–1423).

Nieuwenhuis, C., & Cremers, D. (2013). Spatially varying color distributions for interactive multi-label segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(5), 1234-1247.

Osawa, F., Seki, H., & Kamiya, Y. (2007). Unfolding of massive laundry and classification types by dual manipulator. *JACIII*, *11*(5), 457–463.

Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, *11*(285-296), 23–27.

Qian, C., Yan, Y., & Ai-fang, S. (2008). Review of cloth modeling. In *Education technology and training, 2008. and 2008 international workshop on geoscience and remote sensing. ett and grs 2008. international workshop on* (Vol. 1, pp. 238–241).

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, *1*(3), 244–256.

Ramisa, A., Alenya, G., Moreno-Noguer, F., & Torras, C. (2012). Using depth and appearance features for informed robot grasping of highly wrinkled clothes. In *Robotics and automation (icra), 2012 ieee international conference on* (pp. 1703–1708).

Ren, X., & Malik, J. (2003). Learning a classification model for segmentation. In *Computer vision, 2003. proceedings. ninth ieee international conference on* (pp. 10–17).

Ren, Z., & Shakhnarovich, G. (2013). Image segmentation by cascaded region agglomeration. In *Computer vision and pattern recognition (cvpr), 2013 ieee conference on* (pp. 2011–2018).

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. In *Acm transactions on graphics (tog)* (Vol. 23, pp. 309–314).

Stria, J., Průša, D., & Hlaváč, V. (2014). Polygonal models for clothing. In *Advances in autonomous robotics systems* (pp. 173–184). Springer.

Stria, J., Průša, D., Hlaváč, V., Wagner, L., Petrik, V., Krsek, P., & Smutny, V. (2014). Garment perception and its folding using a dual-arm robot. In *Intelligent robots and systems (iros 2014), 2014 ieee/rsj international conference on* (pp. 61–67).

Sun, L., Aragon-Camarasa, G., Rogers, S., & Siebert, J. (2015, May). Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In *Robotics and automation (icra), 2015 ieee international conference on* (p. 185-192). doi: 10.1109/ICRA.2015.7138998

Suzuki, S., et al. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, *30*(1), 32–46.

Teh, C.-H., & Chin, R. T. (1989). On the detection of dominant points on digital curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *11*(8), 859–872.

Vedaldi, A., & Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In *Computer vision–eccv 2008* (pp. 705–718). Springer.

Willimon, B., Birchfield, S., & Walker, I. D. (2011). Model for unfolding laundry using interactive perception. In *Intelligent robots and systems (iros 2011), 2011 ieee/rsj international conference on* (pp. 4871–4876).