Enabling Garment-Agnostic Laundry Tasks For A Robot Household Companion

David Estevez*, Juan G. Victores*, Raul Fernandez-Fernandez*, Carlos Balaguer*

RoboticsLab Research Group, Universidad Carlos III de Madrid, Leganés, Spain

Abstract

Domestic chores, such as laundry tasks, are dull and repetitive. These tasks consume a significant amount of daily time, and are however unavoidable. Additionally, a great portion of elder and disabled people require help to perform them due to lack of mobility. In this work we present advances towards a Robot Household Companion (RHC), focusing on the performance of two particular laundry tasks: unfolding and ironing garments. Unfolding is required to recognize the garment prior to any later folding operation. For unfolding, we apply an interactive algorithm based on the analysis of a colored 3D reconstruction of the garment. Regions are clustered based on height, and a *bumpiness* value is computed to determine the most suitable pick and place points to unfold the overlapping region. For ironing, a custom Wrinkleness Local Descriptor (WiLD) descriptor is applied to a 3D reconstruction to find the most significant wrinkles in the garment. These wrinkles are then ironed using an iterative path-following control algorithm that regulates the amount of pressure exerted on the garment. Both algorithms focus on the feasibility of a physical implementation in real unmodified environments. A set of experiments to validate the algorithms have been performed using a full-sized humanoid robot.

Keywords: robotics, computer vision, ironing, garments, deformable objects, force/torque control
2017 MSC: 68T40, 68T45

1. Introduction

Domestic tasks, such as doing the laundry or cooking, are tasks in which people spend can over 20 hours of their weekly time on average [1]. They are indispensable, and can span over even longer time ranges, depending on the number of inhabitants of a household. It is certain that some of these tasks, such as cooking, are considered pleasurable for a portion of the population. However, they are globally referred to as unpaid household work, as paid worker substitutes exist. Moreover, among these tasks, some are despised by the majority of the human population[1], such as making the bed, ironing or doing laundry. Yet another issue to take into account is the fact that some collectives, such as elder people and disabled people, require help to perform these tasks, due to factors such as lack of mobility and coordination.

A proof of the high demand for automated domestic task systems is the existence of the recent European projects devoted to this topic. One example is the CloPeMa[2] EU-FP7 research project. The objective of this project, which took place from 2012 to 2015, was to advance the state of the art in perception and manipulation of fabric, textiles and garments. The I-DRESS[3] project (2015-2018) is a CHIST-ERA project focused on assisting people with dressing. The target population are disabled users or high-risk health-care workers, that require limited physical contact with the garments to reduce the contamination risk.

Devices that have been developed to specifically perform some of these daily tasks currently exist (washing machines, dryers, etc). However, as the number of tasks to automate grows, owning one machine per household task becomes unfeasible, due to economical restrictions and physical space limitations at home. Therefore, the path towards implementing feasible solutions must be through using adaptive automated solutions that can perform several of these tasks, if not all. Hence we arrive to the conclusion that the most suitable solution is to aim at robotic systems which require a minimum setup. Ultimately, there is a need for the development of a completely integrated Robot Household Companion (RHC) that can perform daily life, creative, and assistive tasks.

While the idea of a domestic robot companion has been present on the wish list of humanity for decades [2], several challenges arise due to the complexity of the tasks to accomplish. These challenges have limited the deployment of such solutions in real domestic environments for the time being. The RHC is intended to work in human environments, and sometimes in close collaboration with

*Corresponding author
Email address: destevez@ing.uc3m.es (David Estevez)

[1] https://yougov.co.uk/news/2017/02/21/doing-ironing-household-chore-brits-hate-most/ (last accessed: 18-05-2017)

[2] http://www.clopema.eu (last accessed: 18-05-2017)

[3] https://i-dress-project.eu/ (last accessed: 18-05-2017)

people. Human environments, and specially domestic environments, are adapted to the morphology of the humans who inhabit them. Typically, they have obstacles such as tables, chairs or even objects on the floor which complicate navigation and maneuverability. Furthermore, these obstacles are not fixed, their position change along the day due to the different human activities being executed in the environment.

To perform laundry tasks, the system must be able to work with garments. Garment perception is a difficult task, since most of the garments we use on a daily basis can be described as thin deformable objects. Deformability and lack of rigidity allow a single garment to adopt an almost infinite number of different poses, all of them belonging to the same garment. This fact, added to the variety of shapes, textures, and colors, and the possibility of occlusions between garments or self-occlusions, makes recognizing the garment's category and pose a challenge for automated systems.

As challenging as garment perception is garment manipulation. Not being rigid objects increases the difficulty of manipulation tasks, as the shape of the garment, and therefore the sensed shape and weight distribution change during manipulation. Even though attempts have been made to model garments with links and springs models [3], garments show a somewhat chaotic behavior, and it is extremely difficult to precisely predict the pose of a garment after a manipulation operation.

There is an overwhelming number of different daily tasks that can be automated. Even if we aim only at laundry-related tasks, the amount of available tasks is so big that it becomes intractable. Therefore, this work intends to present a first approach towards the Robot Household Companion, and focuses on just two of these tasks: unfolding and ironing garments.

Humanoid robots seem to be the ideal candidates for the application, as their anthropomorphism allows them to have dexterity and locomotion capabilities similar to that of humans. More precisely, the use of a humanoid robot enables addressing the three main challenges described above:

- Due to their human-based morphology, humanoid robots are capable of using different human tools for different tasks, eliminating the need for specialized robotic tools. Even if the selected tasks make extensive use of the robot upper limbs for manipulation, legs are still a valuable asset, since they allow navigating and maneuvering through human environments avoiding obstacles, a requirement for a robot working in a domestic environment.

- Humanoid robot morphology enables obtaining several views of the same deformable object, increasing the possibilities of a correct recognition of the garment category and pose.

- Humanoid robots have two manipulator arms, which increase the range of different manipulation actions that can be performed with garments. They additionally allow the robot to emulate human behaviors when manipulating garments.

This papers integrates our previous work on unfolding and ironing, published in conference papers (see [4], [5] and [6]), into a unified Robot Household Companion. Both algorithms focus on a model-less approach, in which the robot has no prior model of the garments to work with, and on feasibility of the algorithm in a real-world domestic environment, by using unmodified human tools and existing environment illumination. In the case of the unfolding algorithm, in addition to the humanoid robot, industrial robots were used as a platform to perform extensive testing, as well as for validation of the algorithm on different robotic platforms. The previous work have been extended with further experiments, more detailed results in terms of performance and timing of the algorithms, along with a section discussing the benefits of some design decisions over others, justifying the final design. Figure 1 depicts the experimental setup for the ironing task using no special lighting conditions, an arbitrary garment, and an unmodified iron and ironing board with the full-size humanoid robot TEO.

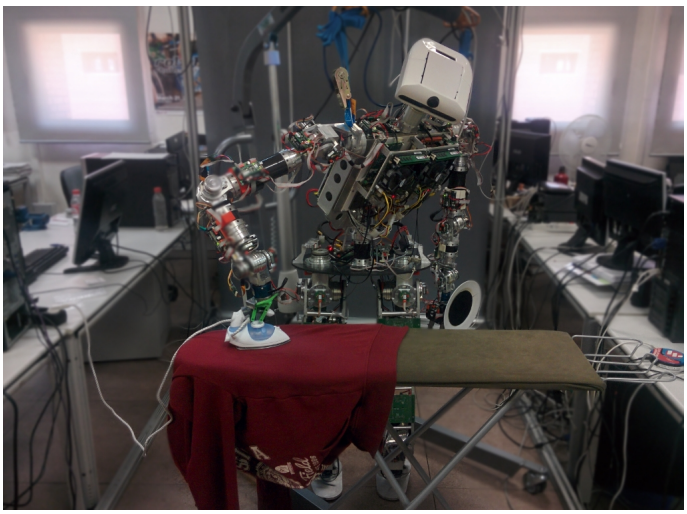


Figure 1: Experimental setup for a full-sized Robot Household Companion using unmodified human tools for ironing.

2. State of the Art

To build a robotic assistant for laundry tasks, a robot must be able to perform all the tasks of a typical laundry pipeline. These tasks involve the following steps: picking up garments from the washing machine, unfolding them to recognize the garment, ironing the garment and, finally, folding the garment.

The first step of this pipeline is to pick a garment from a pile of garments. There are several methods to select

the best pick strategy [7]. For instance, one can use wrinkleness descriptors based on depth data to select the pick point [8].

The second step, after a garment has been picked up, is to unfold it. Osawa et al. propose a method for garment unfolding using a dual-arm manipulator [9]. A single garment is picked up with one manipulator arm and its lowest point is located through the rotation of the garment. Unfolding is achieved by iterative regrasping and placement of the garment over a flat surface. Cusumano-Towner et al. employ a simulated cloth to train a Hidden Markov Model (HMM), which estimates the identity of the garment and its current pose [10]. Through this HMM, and after a series of manipulations and observations, the robot can estimate the steps required to arrive to a desired configuration. Using depth data, Willimon et al. locate the most suitable points and orientations for the robot to unfold the garment by detecting certain features, such as peak regions or corners [11]. Li et al. label a simulated mesh model of the garment with the best grasping points [12]. This mesh model is later matched with the garment, which is unfolded though iterative regrasping with a two-arm manipulator. Triantafyllou et al. use a template dataset to unfold single folds [13]. The image obtained with their cameras is compared to these templates using shape matching to determine the fold edge and unfold the garment.

As an intermediate step after the garment has been unfolded, the garment category is typically recognized to ease subsequent steps. One of the first approaches by Kita et al. compares 3D garment data with candidate shapes generated through simulation [14]. The most similar are selected and further deformed to match the 3D data, selecting the one that better fits the sensor data. Li et al. use sparse coding and 3D shape matching to perform garment recognition [15]. A codebook is constructed with simulated garments in different hanging poses, and these models are compared with the 3D sensor data using a custom descriptor. A more interactive approach is proposed by Doumanoglou et al., which use Active Random Forests to recognize the garments using depth images [16]. This method allows the robot to propose different hypotheses and perform actions such as changing the viewpoint to disambiguate them. Stria et al. propose the use of polygon-based 2D models to recognize fully spread garments based on characteristic landmarks of the garment contour [17].

The ironing step typically comes before the garment can be folded and stored. Some large soft wrinkles can be eliminated by pulling the garment extremes. Sun et al. propose a method to detect these large soft wrinkles by a B-spline simplification and differential geometry analysis of the garment surface [18]. This surface is obtained through a high-resolution active stereo system composed by two SLR cameras. If ironing is required, the garment boundaries need to be identified [19] to place the garment over the ironing board. The first works on robotic ironing focused on analyzing and characterizing the ironing process and ironing trajectories as performed by humans [20]. The study was extended in [21], using the obtained ironing trajectories, orientation and motion for the automated generation of an ironing plan based on different ironing regions. A later approach by Kormushev et al. performs the analysis learning of the human ironing trajectories directly through the robot, using robot imitation to extract the knowledge of how to perform the ironing task from human demonstrations [22]. In this case, the ironing position profile is learned by the robot by kinesthetic teaching. Then, a haptic device is used to teach the robot how much force it needs to apply when performing the ironing operation. In the previous approaches, the ironing operation is performed in open loop, as the system does not take into account the state of the garment regarding the amount, position or size of the wrinkles present. Li et al. use a monocular camera and two manually-controlled light sources to reconstruct the fold marks (wrinkles of negligible volume) present on the garment through a diffuse reflection model, while wrinkles are detected through an RGB-D sensor [23]. The approach does not use force feedback in the control loop, so the ironing operation is performed with the garment placed over a thick piece of foam to achieve a certain degree of passive compliance.

In the final step, once the garment has been ironed, it can be folded and stored. The folding operation is typically based on the sequences used by humans when folding garments. To recognize the different parts of the garment in order to follow those sequences, Miller et al. propose in [24] a method based on parameterized shapes. Each garment category is described with a different set of parameters and, using images from the actual garment, the parameters are fit to describe the garment.

In our work, the methods and algorithms for unfolding and ironing focus on being model-less and feasible on a real-world environment. For unfolding, we propose a method based on using depth information to determine the fold edge and unfolding direction [4] [5]. Our approach detects the overlapped regions of the garment by clustering regions of the image based on their height. For ironing, we removed the need for a controlled light environment using only data obtained with an RGB-D sensor [6]. A custom descriptor was used to detect the wrinkled regions and to direct the ironing operations to those regions. To actively regulate the amount of force exerted to the working surface, the ironing position/velocity control policy included force feedback regarding the axis perpendicular to the surface, removing the need for passive compliance. With respect to the other methods present in the literature, our work provides the following contributions:

- Our method allows the robot to unfold and iron a garment without any prior model of the garment (i.e. no database of garments, no parametric model of the whole garment).
- Our method can be applied to a real domestic environment, as it does not require controlled illumi-

nation or any other modification of the surrounding environment.

- Additional modifications of the tools or working surface, such as adding a foam to the ironing board to compensate lack of force control, have also been avoided.

3. Model-less Unfolding Algorithm

This section describes the garment-agnostic unfolding algorithm for unfolding clothes using only a 3D reconstruction of a garment using a depth sensor, with no need of color information. The main advantage of this algorithm with respect to the previously described methods [9, 10, 11, 12, 13] is that it eliminates the need for a prelearned model of the garment, or any other prior knowledge about the shape of the garment we intend to unfold. This allows our algorithm to work with any kind of garment or textile article, even if it has an uncommon or irregular shape.

The assumption for this algorithm is that the garment has already been laid flat over a flat surface, either by the robot or by a human working with the robot. The simplest way of achieving this with a humanoid robot would be to pick the garment up from any point and, as in other works [9, 10, 16], picking the lowest point with the other arm while hanging, to finally lay it over a flat surface. Once flat, the garment may have several folds that we would like to remove to be able to obtain the garment category from the extended view. The presented algorithm can be applied to remove those folds.

First, the robot scans the garment, obtaining a 3D reconstruction of the garment. Then, the three main stages of the algorithm are applied. In the first stage, the garment is segmented from the background, and the garment contour is approximated. Then, in the next stage, different clusters are computed from the height distribution of the garment, where some clusters correspond to overlapped regions and others represent the underlying part of the garment. Finally, in the last stage, the information about the garment contour and height clusters is used to determine which the best pick and place points are to perform the unfolding operation. This process can be repeated iteratively until the garment is completely unfolded.

3.1. Garment Segmentation

The input of this stage is a high-resolution 3D reconstruction of the garment and workspace. This reconstruction combines different depth views of the garment to obtain a 3D scan with higher resolution than the original depth views, and can be generated using algorithms such as Kinect Fusion [25]. With Kinect Fusion, the reconstructed scene represents the garment with an average resolution of 4 millimeters.

As this scan also includes points from the working surface, a segmentation stage is required to compute which points belong to the garment and which points belong to the background. For this purpose, RANSAC [26] is used to find the working plane, which is then removed from the point cloud. The remaining points are grouped using Euclidean Clustering [27], and the largest cluster is labeled as the garment. This step removes small background clusters that could remain after removing the working plane due to noise or other small elements of the workspace. Once the garment points have been identified, the moments of inertia of those points are computed, and used to obtain the bounding box of the garment. This bounding box, along with the estimated distance between points (or point density of the point cloud) is used to compute a 2D projection of the point cloud, which is used as the height map in the next stage.

The projection is computed as follows. The bounding box is discretized along the local X and Y axes into a set of buckets in the shape of rectangular prisms. The size of the side of the base is dimensioned according to the distance between points in the garment point cloud. For each of these buckets, the point belonging to the bucket with the highest distance to the XY plane local to the bounding box is recorded. With all the recordings, that are distributed in a 2D grid, a 2D projection of the garment can be obtained and used as the height map.

A similar process is used to compute a 2D segmentation mask, but in this case the buckets only record the occupancy of garment points for each bucket. The occupancy is stored as a binary value representing whether the bucket is empty or has any point. All of these buckets, also distributed as a 2D grid, are interpreted as a 2D binary image corresponding to the segmentation mask of the garment. This image can contain some empty pixels due to non-uniformities in the point distribution of the 3D scan, therefore a post-processing step is required. This step consists in applying a morphological closing operation, which involves a dilation operation followed by an erosion operation to remove any impulse noise that could be present in the mask.

Once the mask has been obtained, the garment outline is extracted from the mask using a blob labeling algorithm. This outline is simplified using the Ramer-Douglas-Peucker algorithm [28] to obtain a garment approximated polygon. Each edge of this polygon represents one possible fold candidate. The garment approximated polygon is used in the last stage to compute the fold and most suitable pick and place points. Figure 2 depicts the output of this stage.

3.2. Garment Depth Map Clustering

With the height map and the segmentation mask of the garment already computed, the next stage is to recognize different regions in the height map. These regions, characterized by their height, represent parts of the garment overlaid on top of other parts of the garment, i.e., folds.

Prior to the clustering of the points belonging to these regions, the height map computed in the previous stage

Figure 2: Output of the Segmentation stage. The original RGB image of the garment is shown with the segmentation mask and approximated garment polygon overlaid on top. To be able to see the original image, the mask is shown with a transparency (alpha) of 0.5.

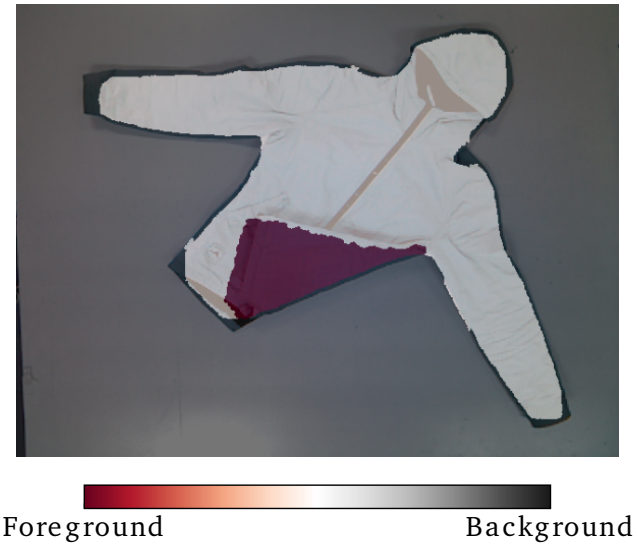


Figure 3: Output of the Clustering stage. The original RGB image is shown with the clustered image overlaid on top. To be able to see the original image, the clustered image is shown with a transparency (alpha) of 0.5. Clusters closer to the foreground are shown in purple, whereas clusters closer to the background are shown in black.

has to be pre-processed. The segmentation mask computed in section 3.1 is then applied to the height map to discard height values not belonging to the garment. The resulting values are then normalized using the depth range (maximum and minimum values) of the height map, and then converted to an 8-bit grayscale image.

The resulting grayscale image is used as input to the clustering algorithm to group the pixels of the image into clusters representing different garment regions. The clustering algorithm used is the Watershed Transform [29]. The Watershed Transform algorithm interprets pixel values as topographical relief, and floods this terrain starting from low regions, called basins, until the different flooded regions meet. The meeting lines, or watershed lines, are interpreted as the contours of the different regions segmented. Each flooded region, then, corresponds to a different cluster.

The height median for each cluster obtained is computed, and then set on each pixel of that cluster, obtaining the clustered depth image $\mathcal{D}(u, v)$, where u and v are the pixel coordinates in the image frame of reference. Using the median instead of the average value for the region makes the algorithm more robust against noise and possible outliers present in the height map. The region with the highest median value is selected for further processing in later stages. The output of this stage can be seen in Figure 3.

3.3. Garment Pick and Place Points

The garment approximated polygon and the different clusters obtained in the previous stages are used as input of this stage, whose objective is to find the most suitable pick and place points by analyzing the input data, as shown in Figure 4. With the pick and place points, a manipulation operation can then be commanded to the robot to perform the actual unfolding of the garment.

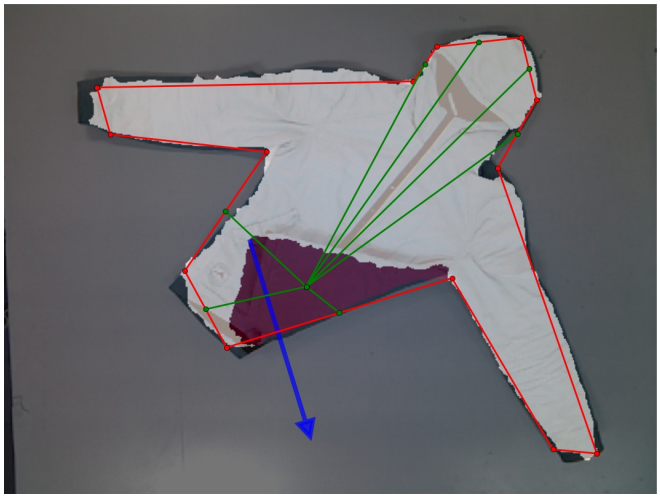


Figure 4: Output of the Pick and Place Points stage. The original RGB image is shown with the clustered image, valid candidate paths and unfolding direction overlaid on top. To be able to see the original image, the clustered image is shown with a transparency (alpha) of 0.5. Valid candidates paths are green and the unfolding direction is represented by a blue arrow.

As mentioned before, each of the edges of the garment approximated polygon represents a candidate to be a fold edge. To determine which of them is actually a fold edge, a set of paths is created, starting from the centroid of the highest region and arriving to the midpoint of each edge of the polygon. Paths intersecting with a segment of the garment outline other than the one used to create the path are considered invalid and discarded. The candidate paths are shown as green segments in Figure 4. The aim

of these paths is to examine the changes in height of the garment along these paths and the discontinuities present within their values. Since the differences in height between the overlapped and underlying regions are assumed to be larger than the differences in height within each of these regions, the changes in height and discontinuities can be exploited to evaluate which path leads to a fold edge. The metric representing these changes and discontinuities along a given path is called *bumpiness* (B):

$$B_{path} = \sum_{i=2}^{m} |s_i - s_{i-1}|, \tag{1}$$

where the *bumpiness* value B_{path} is defined as the accumulated relative difference of a set of m sampled height values s_i computed along a given path. Each of the paths, of length L, is divided in segments of constant length l, and therefore the total number of sampled height values m can be expressed as:

$$m = \left\lfloor \frac{L}{l} \right\rfloor \tag{2}$$

The clustered depth image $\mathcal{D}(u, v)$ is sampled at these m discrete points of each of the n paths, generating n ordered sets $S = \{s_1, ..., s_m\}$, in which each element is computed as follows:

$$s_i = \mathcal{D}(path(i \cdot l)), \quad i = 0, 1, 2, ..., m \ , \tag{3}$$

where *path* is a 2D parametric line ($\mathbb{R} \rightarrow \mathbb{R}^2$) that depends on a single parameter r corresponding to the radial distance from the highest point to a given point of the line:

$$path(r) = [u(r), v(r)] \tag{4}$$

The unfolding direction is determined by the candidate path whose sampled height points have the lowest *bumpiness* (B) level. This path corresponds to the direction with less discontinuities and height variations.

To compute the pick point, the intersection between the selected unfolding direction and the contour of the highest cluster is computed. The intersection of both contains, at least, two points. The pick point is the furthest one from the approximated garment polygon edges, measured along the unfolding direction.

To obtain the place point, the pick point is mirrored using the fold edge as the mirroring axis. The fold edge corresponds to the edge of the garment approximated polygon containing the midpoint used to generate the path candidate selected as the unfolding direction. Using the mirror point reduces the possibility of dragging when unfolding, as it takes the final shape of the garment when spread into account. The pick and place points are shown in Figure 4 as the start and end points of the blue arrow.

Once the pick and place points have been computed in the image frame of reference, obtaining their corresponding 3D points in the robot frame of reference is performed by reverting all the transformations performed to the original point cloud in the segmentation and clustering stages. These points are then sent to the robot to manipulate the garment by executing a conventional pick and place operation.

4. Model-less Ironing Algorithm

This section describes the garment-agnostic ironing algorithm for ironing garments. Existing approaches [23] focus on detecting individual fold marks (wrinkles of negligible volume), and then pressing the iron against each of the individually detected fold marks. These methods demand a multi-sensor approach that requires very fine control on the illumination conditions of the garment, which prevents their application in real domestic environments.

Our approach, on the other hand, is based on observations of humans ironing garments. Instead of looking for individual fold marks to target them, humans tend to iron the whole garment surface while focusing on eliminating large wrinkles or creases to avoid creating new fold marks. In addition it can be assumed that, as wrinkles are regions where the iron has not been applied yet, there is a high probability of finding fold marks in these regions. Therefore, by detecting and ironing the wrinkles in a controlled manner, fold marks can be eliminated. Since only wrinkle detection is critical under this assumption, it is sufficient to work with a colored 3D reconstruction of the working environment. While this limits our method to detecting wrinkles with a relatively strong volumetric signature, it however enables a feasible real world implementation in a domestic setting which has not been achieved to this date with a fold mark detection approach to the best of the authors' knowledge.

Our algorithm consists of two main components: a 3D perception-based ironing path generation and a path-following hybrid control. First, a pre-programmed ironing path is executed from an initial home position to reach a position above the iron board while avoiding self-collision and collision with the board. The robot learns this trajectory by extracting waypoints from a one-shot demonstration with the robot arm set to gravity compensation mode. Once the pre-programmed ironing path has been completed, the path-following hybrid control uses wrinkle information obtained by the 3D perception-based algorithm to remove one wrinkle. To completely remove all the wrinkles, this algorithm can be repeated several times, reducing the amount of wrinkles, and therefore the amount of fold marks with each pass. Therefore, these components are designed to work sequentially and iteratively.

4.1. 3D perception-based ironing path generation

This first component, the 3D perception-based ironing path generation, computes the most suitable ironing path to follow to reduce the total wrinkleness of the garment. It is composed by three different stages. First, the working environment, including the garment and the ironing

board, are reconstructed in 3D with Kinect Fusion [25]. Then, the garment is segmented from the ironing board, and a custom descriptor is computed. This descriptor represents the wrinkleness of the garment. Finally, wrinkles are found with the former descriptor, and the most suitable ironing path is computed and sent to the next component to execute the ironing action.

The process starts with a 3D reconstruction of the working environment, including the garment and the ironing board, shown in Figure 5a. To obtain this reconstruction, an RGB-D sensor is used. Since the resolution of a single depth frame is not sufficient to detect the most subtle wrinkles, the Kinect Fusion algorithm is used, where several views from different poses are continuously integrated in a single 3D reconstruction of the garment, obtaining a higher resolution point cloud. This algorithm requires parallax among views to obtain enough disparity, which can be achieved by translation of the sensor, rotation on more than one axis, or a combination of both. These movements depend on the robot morphology and are therefore platform-specific. From the 3D colored reconstruction, the garment data needs to be extracted. For this purpose, the ironing board surface is located and extracted with RANSAC. From all planes detected with RANSAC, the ironing board surface is extracted using the following prior: the ironing board surface corresponds to the plane whose normal is aligned with the global Z axis, and has the highest Z component of all those planes. The resulting plane contains not only the ironing board, but also the region of the garment laying on top of it, which is the region of interest, as depicted in Figure 5b. To separate it from the ironing board, the points are assumed to be linearly separable based on color. Two sets of points X_G and X_B must fulfill the following requirement:

$$\sum_{i=1}^{n} \theta_i x_i > k \qquad \forall \vec{x} \in X_G \tag{5a}$$

$$\sum_{i=1}^{n} \theta_i x_i < k \qquad \forall \vec{x} \in X_B, \tag{5b}$$

where θ_i with $i \in \{1, \dots, n\}$ are weights, and k is a real number. Hence, the set of n points contained in the ironing board plane can be partitioned into $k = k_G + k_B$ different clusters, where k_G of them belong to the garment and k_B belong to the ironing board. Under a uniform color and a Lambertian reflectance model assumption for both the garment and the ironing board surface, the total number of clusters to be considered is $k = 1+1 = 2$. The assumption of uniform color limits our model to plain garments, but offers a good trade-off between model complexity and the amount of garments to which it can be applied. To improve separability, color is converted from the RGB space to the HSV space. The within-cluster sum of squares can then be iteratively minimized using the following expression for k clusters:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{\vec{x} \in S_i} \left\| \vec{x} - \mu_i \right\|^2, \tag{6}$$

where S is the set of clusters, k is the number of clusters, S_i is the i-th cluster, $\vec{x}$ is the feature vector, and μ_i is the mean of the points in S_i. The feature vector $\vec{x}$ includes both the spatial location of the point (x, y, z) and its HSV color components (h, s, v).

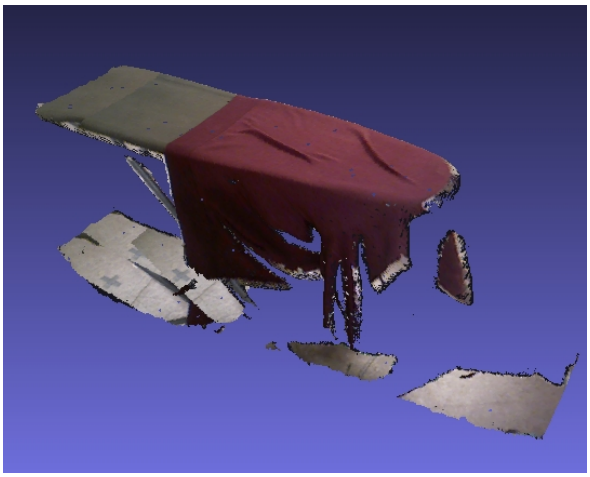
The color clusters obtained are labeled according to the position of their centroids. Since the ironing board has been placed in front of the robot by the user, the orientation of the ironing board is known. Nevertheless, this prior could be easily removed by obtaining its orientation through a simple shape analysis, given the fact that the ironing board has a very characteristic shape. Once the orientation of the ironing board is known, the location of the garment can be deduced from it. Euclidean clustering is then applied to the selected cluster to remove outliers and obtain the filtered garment data. The final result of the segmentation process can be seen in Figure 5c.

In the next stage, a custom Wrinkleness Local Descriptor (WiLD) that is used in the last stage to locate the wrinkles is computed. The WiLD descriptor is defined as:

$$\mathrm{WiLD}(\vec{i}) = \frac{1}{\varphi} \cdot \sum_{\vec{j} \in \Phi} \vec{n}_{\vec{i}} \cdot \vec{n}_{\vec{j}}, \tag{7}$$

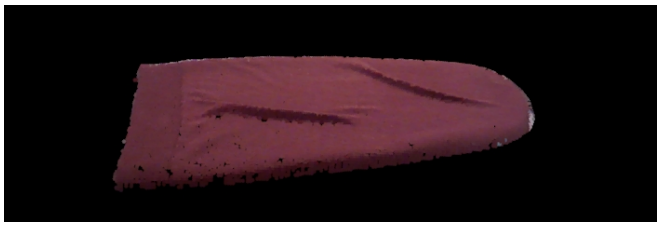
where $\bar{n}_{\vec{i}}$ is the normalized normal vector computed at point $\vec{i}$, $\vec{i}$ denotes the point for which the descriptor is computed, and Φ is the set of φ neighboring points within a radius r. The WiLD descriptor is within the range between 0 and 1, where values close to 0 mean that the point is placed at an edge, and values close to 1 mean the point is placed at a flat region. Wrinkles are not as sharp as edges, therefore, their descriptor values result in intermediate values that depend on the abruptness of the wrinkle. The values for the different points are projected into a 2D image using the garment plane as the projection plane, and z-buffering [30] to manage possible occlusions. A segmentation mask is also generated using the projection, assigning a value of 1 when a pixel is occupied by a garment value and 0 when it corresponds to the background.

The next stage takes as input the WiLD image and the segmentation mask obtained in the previous step. This segmentation mask can present some missing pixels due to the discretization of the point cloud. To fill in these gaps, a closing morphological operation is performed. This operation is composed by a dilation followed by an erosion. In this case, the erosion performed was larger than the previous dilation, with the objective of removing part of the outer border of the garment. Due to the inherent curvature of the underlying ironing board, the outer border will always have some curvature that could be confused with wrinkles by the algorithm, so it is removed. The outer contour of this mask is extracted to be used in later steps.



(a) 3D reconstruction of the working environment

(b) Ironing board top segmented from the 3D reconstruction



(c) Garment segmented from the ironing board

Figure 5: Different steps of the garment segmentation stage.

The segmentation mask is applied to the WiLD image to obtain the pixels belonging to the inner part of the garment surface, and a threshold operation is applied to obtain the wrinkled regions:

$$wrinkle(\vec{x}) = \begin{cases} 1, & \text{if } L_t < \text{WiLD}(\vec{x}) < H_t \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

where L_t and H_t are the lower and higher thresholds.

As an alternative descriptor to our WiLD, the Radius-Based Surface Descriptor (RSD) [31] could be used. This descriptor measures the curvature of a surface at a given point, encoding the radial relationship of the point and its neighborhood. While RSD can provide more information of the curvature of a surface, it is tailored as a general descriptor. On the other hand, WiLD has been developed specifically for wrinkle detection, and therefore it has a better performance on this problem. A comparison between both descriptors is provided in the experiments section (Section 5.2).

From all the candidate regions obtained, the one with the largest area is selected in each iteration. The ironing path is obtained from the selected area shape. To obtain the ironing path, a skeletonization algorithm is first applied to the region, obtaining a set of pixels corresponding to the skeleton of the region. The algorithm used is the Zhang-Suen algorithm [32]. The resulting pixels are then converted to a graph according to their connectivity. The start and end points from this skeleton graph are selected according to the following criteria:

- The start point corresponds to the leaf node of the graph that is closest to the garment contour.
- The end point corresponds to the leaf node of the graph that is furthest from the garment contour.

Once the start and end points have been computed, the ironing path is obtained by performing a depth-first search (DFS) [33] in the graph using the start and end points as the root of the graph and target, respectively. Figure 6 depicts this process.

4.2. Path-following ironing hybrid control

A path-following algorithm implements a hybrid control scheme which enables following the wrinkle path on the ironing board plane, while controlling force on the perpendicular axis. It uses a set of $w = \{w^{(0)}, ..., w^{(N)}\}$ waypoints obtained from the perception path generation as references for performing manipulation for ironing. These waypoints are described by their Cartesian components, i.e. $w^{(i)} = (w_x^{(i)}, w_y^{(i)}, w_z^{(i)})$, that are obtained by a transformation of the path obtained at the end of the previous section from the image reference system to the RGB-D sensor reference system, and then to the robot base reference system. Figure 7 depicts the different reference systems.

For moving the iron to a position above the garment, close and perpendicular to the ironing board surface, an initial trajectory is performed. This trajectory uses a fixed set of waypoints $f = \{f^{(0)}, ..., f^{(N)}\}$. The end position is fixed at a predefined height, perpendicular and on top of the initial waypoint that is received from the perception system. This position is computed by replacing components $f_x^{(N)}$ and $f_y^{(N)}$ by $w_x^{(0)}$ and $w_y^{(0)}$.

The iron is then vertically lowered until contact with the ironing board is detected by the force/torque sensor, when a threshold $F_d = ((F_{dx}, F_{dy}, F_{dz}), (T_{dx}, T_{dy}, T_{dz}))$ is reached, where (F_{dx}, F_{dy}, F_{dz}) and $(T_{dx}, T_{dy}, T_{dz}))$ correspond to the Cartesian components of the force and torque measured by the sensor. The lowering, in joint space, is ultimately position-controlled or velocity-controlled. In position control, a small joint space increment Δq_{cmd}, which is computed given a desired Cartesian increment vector

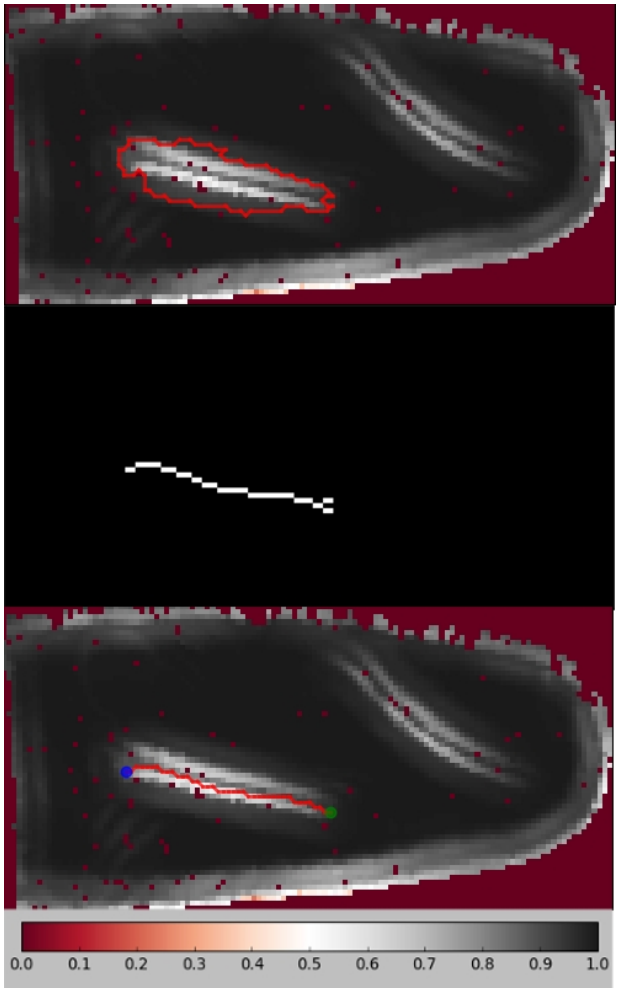


Figure 6: Different steps of the ironing path extraction from the WiLD 2D image. The largest wrinkled region is first selected (top). For this particular example, the leftmost wrinkle spanned $\sim$ 3000 px and the rightmost one $\sim$ 2000 px. The selected region is then skeletonized (middle), and from this skeleton the most suitable ironing path is computed (bottom). The two points, blue and green, represent the start and end points of the ironing path, respectively.

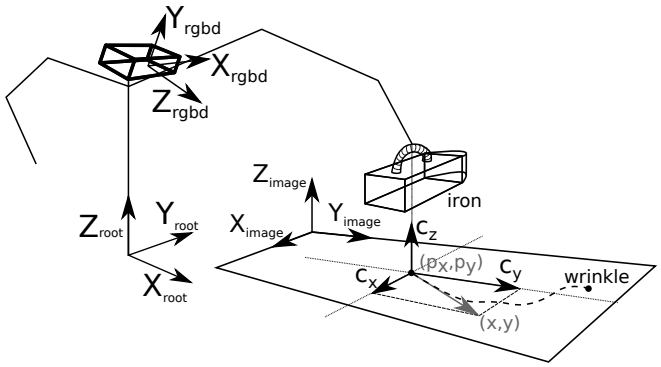


Figure 7: Reference systems for the ironing algorithm, including the robot root frame, the RGB-D sensor frame and the image frame, along with the control vector $\vec{c}$.

$\Delta x_d = (0, 0, -x_{dz})$ and using the Levenberg-Marquardt algorithm [34] to perform inverse kinematics, is commanded. In velocity control, a velocity control loop $\dot{q}_{cmd} = J_A^{\dagger}(q) \cdot \dot{x}_d$ is implemented, where $\dot{q}_{cmd}$ is the joint velocity vector that is commanded, $J_A^{\dagger}(q)$ is a Moore-Penrose pseudoinverse that uses singular value decomposition (SVD) based on householders rotations with updated joint positions q at each iteration, and $\dot{x}_d = (0, 0, -\dot{x}_{dz})$ is the desired Cartesian velocity vector.

To iron the detected wrinkles, once contact has been detected, the ironing path $w = \{w^{(0)}, ..., w^{(N)}\}$ received from the vision system must be followed. The path-following hybrid control is governed by the control vector $\vec{c}^{(\tau)}$:

$$\vec{c}^{(\tau)} = \begin{cases} c_x^{(\tau)} = c_t \cdot cos(atan2((w_y^{(i)} - y^{(\tau)}), (w_x^{(i)} - x^{(\tau)}))) \\ c_y^{(\tau)} = c_t \cdot sin(atan2((w_y^{(i)} - y^{(\tau)}), (w_x^{(i)} - x^{(\tau)}))) \\ c_z^{(\tau)} = c_z^{(\tau-1)} + K_f \cdot (F_{dz} - F_z^{(\tau)}) \end{cases} , \quad (9)$$

where τ is the time step, $(c_x^{(\tau)}, c_y^{(\tau)}, c_z^{(\tau)})$ are the Cartesian components of $\vec{c}^{(\tau)}$, c_t is a hand-crafted constant that corresponds to the desired tangential component of $\vec{c}$, the Cartesian components of the instantaneous position are $x^{(\tau)}$ and $y^{(\tau)}$, and K_f is the user-tuned proportional gain of the force control on the vertical axis. Figure 7 shows the control vector $\vec{c}$ in the context of the wrinkle-following trajectory.

In the joint space, control in this stage may also be performed in position or in velocity mode. In position mode, the control vector $\vec{c}$ may be used to as Δx_d to compute a small commanded joint increment Δq_{cmd} through inverse kinematics as previously. In velocity mode, the differential form $\dot{q}_{cmd}$ may be computed through the premultiplication of $J_A^{\dagger}(q)$ on $\vec{c}$ used as $\dot{x}_d$.

The final vertical ascent stage is similar to the lowering stage, where $\Delta x_d = (0, 0, +x_{dz})$ is used for joint space position control through inverse kinematics, or $\dot{x}_d = (0, 0, +\dot{x}_{dz})$ is used for joint space velocity control through differential inverse kinematics. These commands are performed for a given amount of time or distance.

5. Experiments and Results

To test the concept of the Robot Household Companion, a set of experiments were performed for each of the tasks being evaluated.

5.1. Garment Unfolding Experiments

In our previous work, two versions of our algorithm, the original one [4] and the improved one, described in this article, were compared [5]. To increase the throughput of the experiments, they were performed with an industrial manipulator as platform, although the algorithm has also been validated on our full-body humanoid robot called TEO [35]. The experimental evaluation presented in our previous work has been extended with a larger test

set for a total of 100 additional trials with 5 garment categories.

The robot used in the unfolding experiments was an ABB IRB 2400 industrial manipulator robot. This robot was equipped with a custom, 3D printed gripper with EVA foam in the gripper claws to reduce slippage. The gripper was actuated pneumatically. In addition, an ASUS Xtion PRO LIVE RGB-D sensor was attached to the tool to obtain depth data from the scene. The industrial manipulator controller is running a server modified from the open_abb [4] controller that receives the target points and orientations from our main PC.

For each experiment, the garments were placed over a flat hard white surface. Originally, a total of 6 garments for each category were used. The garment categories being evaluated were: Skirt, Jacket, Pants, Polo, Robe and Hoodie. The garment were laid flat over the working surface manually, and then folded to obtain either one or two folds, depending on the experiment. For the latest set of experiments, due to time constraints, only 5 of the previous categories were tested: Skirt, Jacket, Pants, Polo, Robe.

For the evaluation of our previous approach, single 640x480 px depth and RGB images were obtained per garment sample from the ASUS Xtion sensor as input for the algorithm, all captured from a bird's eye perspective. For the current approach, the industrial manipulator was used to rotate the ASUS Xtion camera around the garment, following a circular trajectory and obtaining different views of the garment. Two circular trajectories were performed, a higher one with an average camera inclination of 60° with respect to the table normal, and a lower one, with an average inclination of 30°. These views were integrated by the Kinect Fusion algorithm to obtain a single 3D reconstruction of the garment. As more points are used, the total resolution achieved increases, providing more resolution than a single depth frame. For these experiments, PCL's [36] implementation of Kinect Fusion was used[5]. Once the garment 3D reconstruction had been obtained, the data was used as input for the algorithm, computing the segmentation mask, clustering of the different overlapped regions and the most suitable pick and place points. Those points were then converted from the sensor frame of reference to the robot's root frame of reference, and provided to the robot to perform the unfolding manipulation operation.

Our previous work included a total of 5 trials performed for each of the garments, 3 of them with different single folds, and 2 of them with different double folds. Figure 8 depicts the final output of the improved algorithm and the computed unfolding directions for 2 of the algorithm runs of each of the 6 categories. These results have been extended through further experimental validation, with 20 additional trials per garment category for 5 garment categories, with a total of 100 trials. For each category, 10 of the 20 trials were performed with a single fold, and the remaining ones with 2 folds. For each of the unfolding operations, the robot required an average of 63 ± 1 s for the 3D garment reconstruction and an average of 28.5 ± 6.8 s to perform the actual pick and place operation.

Table 1 provides a stage by stage analysis regarding the experiments presented in our previous work, reporting an increase of performance of the improved version of 20 % with respect to the original approach, yielding an overall success rate of 60 %. As some failures in previous stages of the algorithm (e.g. garment segmentation) could lead nevertheless to a successful clustering and unfolding, each stage of the trial was classified as a success or failure to obtain a more accurate description of the performance of the algorithm. Table 2 provides a stage by stage analysis regarding the experiments with 20 trials for each of the 5 garment categories.

Both tables show a consistent overall performance was obtained for both sets of experiments, slightly decreasing in the clustering stage and increasing for the remaining of the stages. Some failure modes can be observed, for instance, in the case of very thin garments such as the robe or the polo, where RANSAC misinterprets part of the garment as part of the background table. In contrast, previous failure modes occurring with darker garments such as the hoodie are no longer present, as the improved algorithm is color-independent. In the case of the clustering stage, the most common failure mode is caused by single overlapped regions being divided into several clusters, causing unstable bumpiness values. Split clusters can also cause the selected cluster edge to not correspond to the edge of the whole overlapped region, but to a smaller region within it. If this inner edge is used to compute a pick point, it is impossible for the robot to fully unfold the region, as the point would lay below the unfolded region. Therefore a folded region of size equal to the distance between the pick point and the region edge is left after the manipulation operation. To address these issues and improve the overall performance of the algorithm, some discussion is provided in Section 6.

5.2. Garment Ironing Experiments

The experimental setup for the ironing algorithm is intended to simulate a typical domestic environment. A real ironing table was fit with real garments, and a real, standard iron was used to iron the garments. This iron was installed in our robot using custom 3D printed parts. The robot used for the experiments is a full-body humanoid robot called TEO [35], designed and built at the Robotics Lab of Universidad Carlos III de Madrid. As the focus of this set of experiments was ironing, not grasping, one of the hands of the robot was removed to install the 3D printed parts holding the iron. The humanoid robot is equipped with JR3 6D force/torque sensors to obtain the force/torque feedback required for the path-following hybrid ironing control. For visual feedback, the robot fea-

[4] `https://github.com/robotics/open_abb`

[5] `http://pointclouds.org/documentation/tutorials/using_kinfu_large_scale.php`

Table 1: Results analysis of the Unfolding Algorithm (5 trials, 6 categories), per stage and garment category, expressed in percentage (%)

Stage / Category	Skirt	Jacket	Pants	Polo	Robe	Hoodie	All
Original approach							
Segmentation	100	100	80	100	100	20	**83.3**
Clustering	80	60	60	80	60	0	**56.7**
Pick & Place Points	60	40	20	80	40	0	**40**
Improved approach							
Segmentation	100	100	100	60	60	100	**86.7**
Clustering	80	80	80	80	80	80	**80**
Pick & Place Points	60	60	60	40	80	60	**60**

Table 2: Results analysis of the Unfolding Algorithm (20 trials, 5 categories), per stage and garment category, expressed in percentage (%)

Stage / Category	Skirt	Jacket	Pants	Polo	Robe	All
Improved approach (extended trials)						
Segmentation	100	100	95	100	100	**99**
Clustering	85	60	63.2	85	75	**73.6**
Pick & Place Points	70.6	75	33.3	94.1	78.6	**70.3**

tures an ASUS Xtion PRO LIVE RGB-D sensor. Similarly to the previous unfolding experiments, this sensor obtains a 3D reconstruction of the working scene using PCL's implementation of the Kinect Fusion algorithm. As the range of motion for the robot head is more limited than the industrial manipulation used for unfolding, the pan and tilt degrees of freedom of the robot head are used to move the sensor to achieve sufficient disparity to generate the 3D reconstruction. Once the scene is scanned, the 3D reconstruction obtained is fed into our algorithm to compute the ironing path points. As these points are computed in the image frame of reference, they have to be converted to the robot frame of reference before any manipulation operation can be performed.

For the evaluation of the ironing algorithm, two sets of experiments were designed, one for the evaluation of the perception algorithm, and other for the evaluation of the whole algorithm. The parameters used in these experiments were the following: 0.02 for the RANSAC threshold, 0.02 for the normal estimation radius, 0.03 for the WiLD neighborhood radius, 0.95 for the WiLD upper threshold, 0.4 for the WiLD lower threshold, 11 for the erosion structuring element size.

The first set of experiments evaluates the accuracy of the perception algorithm to detect wrinkled regions. For this purpose, 10 trials were performed, 5 with garments with 1 wrinkle and 5 with garments presenting 2 wrinkles. As only the perception component was being evaluated in these experiments, only the 3D reconstruction and posterior wrinkle extraction were performed. The performance of our WiLD descriptor was compared against the performance of the Radius-based surface Descriptor (RSD), using the Jaccard similarity index (JSI) as the metric:

$$JSI(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

To compute the JSI metric, the results from the WiLD and RSD evaluations were compared with hand-labeled ground truth of the 3D reconstruction. WiLD and RSD extracted regions are compared against a hand-labeled ground truth of the scan using the JSI metric. Computation times for both descriptors were also computed, using a machine with an Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz processor and NVidia GeForce GTX 960 graphics card with PCL-1.7 over an Ubuntu GNU/Linux distribution. Table 3 depicts the results of this set of experiments. On average, WiLD presents a 25 % better JSI compared to the RSD, while performing over 40 % faster. The assumption regarding precision in terms of the JSI metric is that WiLD has been developed specifically for wrinkles and does not attempt to be a general descriptor as is the case of RSD. The complexity of both algorithms depends non-linearly on the size of the input to process, which is in the order of millions of points. Because of that, the measured times for both methods lie in a 30 s to 60 s interval, depending on the method. Still, WiLD enables much faster processing through a simple and multi-threaded implementation, which could be further improved by using GPUs instead of CPU multithreading.

The performance of both components of our algorithm are evaluated by the second set of experiments, including the actual manipulation of the iron to perform the computed ironing paths. For the manipulation control, the values of the F_d parameter used are ((0,0,-60),(0,25,0)), with the force measured in N and the torque in $N \cdot m$.

The garment is placed manually and mostly flat over

Table 3: Results of the Perception Algorithm

Experiment		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Mean
RSD	JSI (%)	36.67	32.51	44.90	30.75	41.12	42.27	48.84	38.30	31.07	34.60	38.10
	Time (s)	52.12	50.2347	52.46	52.37	53.24	50.30	49.72	50.42	49.68	51.60	51.22
WiLD	JSI (%)	68.41	60.98	67.39	59.65	61.94	68.64	64.02	65.56	67.03	60.64	64.43
	Time (s)	36.07	34.51	36.69	36.62	38.06	34.88	34.88	35.39	34.23	35.39	35.67

the ironing board for each trial. Two wrinkles are created randomly over the garment. This ensures similar initial conditions for all the trials, in terms of garment state as well as the same number of wrinkles present in all the trials. The algorithm is executed iteratively until the wrinkleness value obtained is sufficiently close to 0 to be negligible. This wrinkleness value is the metric used to evaluate the performance of the algorithm, as well as the progress towards a completely ironed garment, and it is computed as follows:

$$\text{wrinkleness} = \frac{\sum_{\vec{x} \in G} wrinkle(\vec{x})}{|G|}, \qquad (11)$$

where G is the set of points belonging to the garment and $|G|$ represents the cardinality of that set. The number of iterations required, as well as the value of the wrinkleness metric for each iteration are recorded. The time elapsed in each iteration is also recorded. A total of 5 experiments were performed, whose results are shown in Table 4. Note that the times shown only measure the elapsed time per ironing manipulation operation. While intermediate wrinkleness vary for each iteration, convergence to zero wrinkles on only 2 iterations is a very satisfactory result.

This value of wrinkleness is tracked for each iteration and trial, and shown in Figure 9. Figure 10 shows a sequence of frames which depicts one of the ironing operations that was performed during these experiments.

6. Discussion

The results for both tasks show that the algorithms for the unfolding and ironing tasks offer very promising results. For unfolding, the results obtained with the most recent method increased the performance of our previous approach in up to 20 %. This improvement is mainly due to a better segmentation based only on depth data. Still, some garments may yield poor results due to an incorrect segmentation when the thickness of the garment is close to the maximum resolution of the 3D reconstruction. In these cases, fusing RGB data with depth data might increase the performance of the algorithm, but it is not trivial how to perform the data fusion in such a way that the algorithm remains invariant to changes in illumination and suitable for colored or patterned garments in real world conditions. Thin garments may also cause the clustering algorithm to misinterpret overlapped regions, which in most cases can be solved with a fine-tuning of the clustering parameters depending on the garment material or thickness. Other cases are limited by the resolution of the RGB-D sensor and 3D reconstruction method and can only be improved with better hardware or reconstruction algorithms, which are out of the scope of this work. Manipulation of garments is another aspect limited by hardware, as the grippers currently installed in our robots are designed for generic manipulation tasks, and do not provide any kind of tactile feedback or, for some garments, a sufficient amount of gripper-garment friction. Alternatively, the simple gripper could have been substituted by a more advanced robotic hand such as [37] or [38], but it would increase the complexity of the control and would require a more sofisticated grasping point and pose selection.

Ironing is a challenging task to automate due to the great variety of shapes, textures, materials and, especially, decorative elements that have to be ironed in a particular way or even avoided. This work focuses on offering a model-less approach to ironing, and therefore those garment elements that require specific ironing patterns, such as buttons, zippers, pockets, collars, etc are not detected, as that would require a previous model of the garment, or at least a model of the decorative elements. Since garment segmentation is based on color, the algorithm is currently limited to garments and ironing boards with a single, uniform color, and sufficiently distinct from one another. One may argue that the current clustering approach could be extended to detect two clusters based on texture instead of color, but in practice this poses several challenges, especially for garments that have more than one printed pattern or texture, which would hinder separating these patterns in just two clusters. As with the unfolding algorithm, fusing RGB data with depth data might improve the performance of the segmentation algorithm, but for thin garments the resolution of the 3D reconstruction would have to be improved, which is out of the scope of this work.

Our algorithm does not target individual fold marks (wrinkles of negligible volume), and instead removes wrinkles under the assumption of them being regions that have not been ironed yet, and therefore with a high probablity of fold marks being present. For this reason the limited resolution of the 3D reconstruction, which only allows the algorithm to detect wrinkles and bumps with a relatively strong volumetric signature, is enough for the current formulation of the algorithm. Even if our algorithm could be improved by detecting fold marks, that would require

Table 4: Results of the Ironing Algorithm

Experiment	#1	#2	#3	#4	#5
Number of iterations	2	2	2	2	2
Avg. time (s) / iteration	71.827	67.803	72.561	75.684	63.684

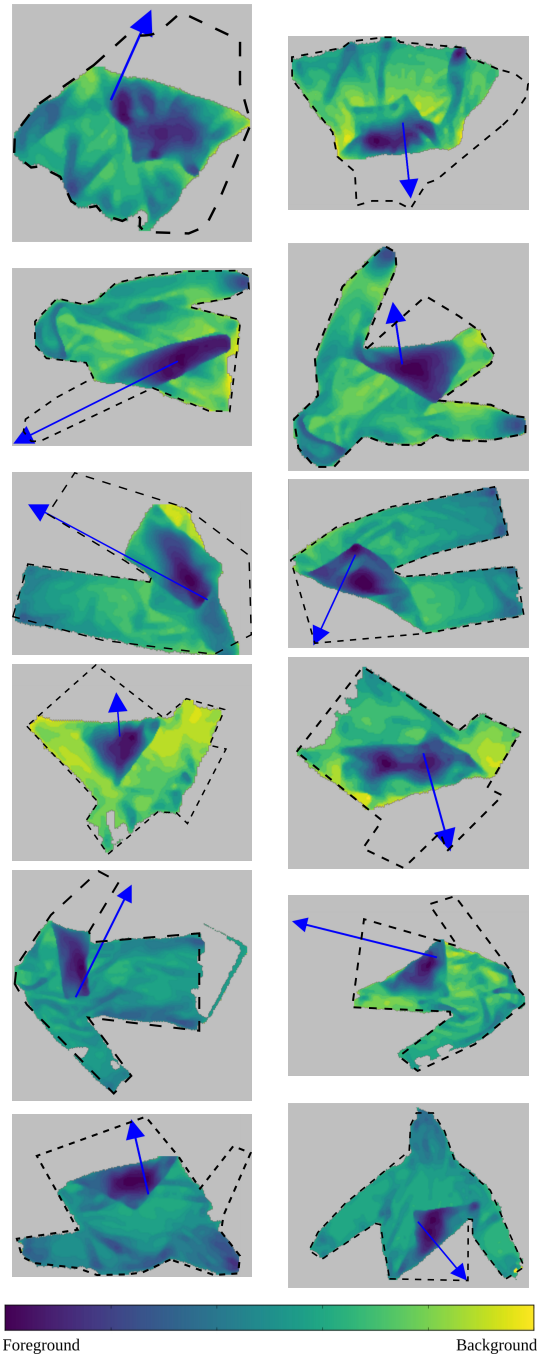


Figure 8: Computed unfolding directions overlaid on top of the corresponding garment height map. Each row includes the output corresponding to 2 of the 5 algorithm runs for each of the 6 garment categories considered: Skirt, Jacket, Pants, Polo, Robe, and Hoodie. The unfolded garment outline has been added by hand to each figure for illustrative purposes.

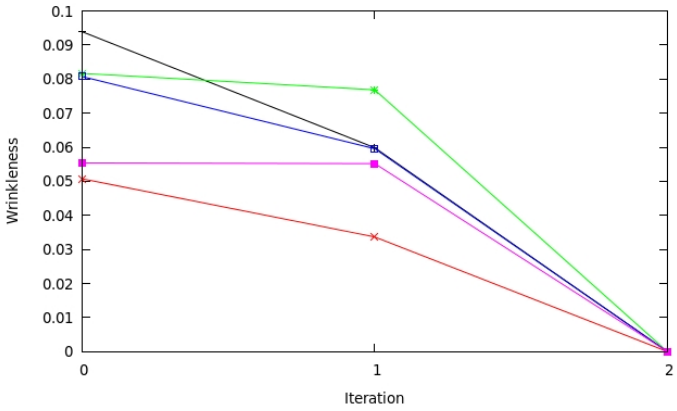


Figure 9: Wrinkleness on each experiment trial, from iteration 0 (initial wrinkleness) to iteration 2. No trial required more than 2 iterations to achieve zero wrinkleness (i.e. no wrinkles between the WiLD threshold could be found).

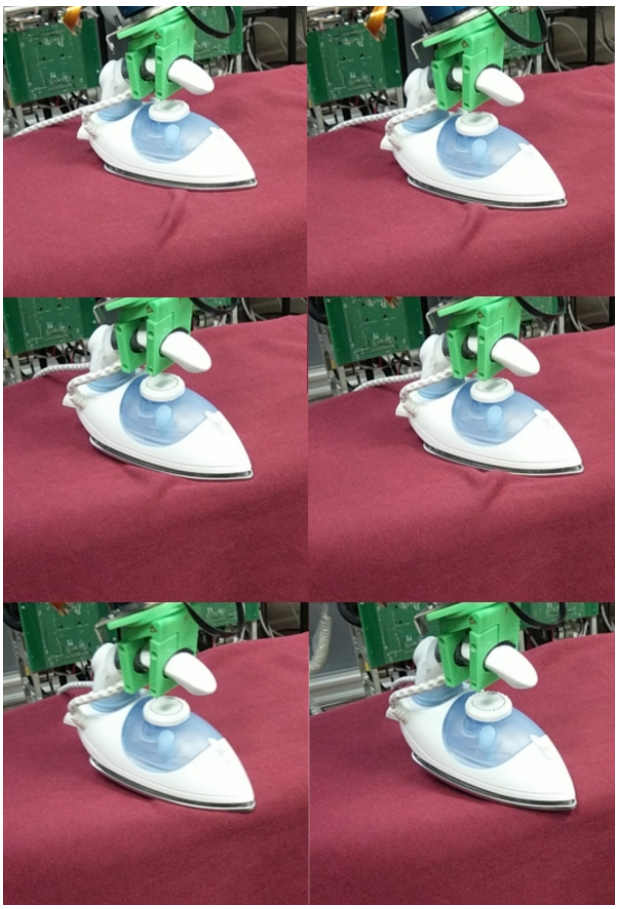


Figure 10: Several frames of an ironing operation executed as part of an experiment.

at least a sufficiently high resolution RGB image of the garment and, based on the work of other authors [23], unrealistic setups requiring controlled external light sources, which defeated our goal of a feasible implementation on a real-world domestic scenario. To the best of our knowledge, a feasible real-world domestic implementation of a fold mark-detection based algorithm has not been achieved yet in any other published work, which is proof of the high complexity of the problem.

For ironing manipulation, position and velocity control have been tested with force/torque feedback. Velocity control provides mechanical benefits for the robot and task as it reduces the jerk of the movements. However, Cartesian space velocities must always be bounded in order to avoid excessive velocities in the joint space, which is a situation that may occur when the determinant of the updated Jacobian is near zero. An additional enhancement could be enabled through the implementation of a torque-controlled scheme, which would be achieved using the transposed Jacobian matrix J^T, providing active compliance as an extra enhancement. However, functional torque-control schemes require precise dynamic models of robotic platforms, which are not usually available in practice.

Finally, the presented work is focused only on unfolding and ironing, but to achieve a Robot Household Companion attractive enough for users to include it in their homes, other laundry-related tasks, such as folding, should be added and integrated to obtain a complete laundry pipeline that includes garment manipulation, unfolding, ironing and folding.

7. Conclusions

This work presents the first developments towards a Robot Household Companion able to perform laundry-related tasks in a domestic environment. For this purpose, we propose methods to tackle two of these tasks: unfolding and ironing. For garment unfolding, we present a model-free garment-agnostic algorithm based on depth data that improves existing work. The limitations overcame by this approach include color-dependent segmentation and arbitrary selection of pick and place points. Several depth frames from different poses are integrated using Kinect Fusion to find the overlapped regions, making our algorithm independent of colors and patterns typically present in garments and working surfaces. In addition, our method only requires one arm for manipulation, whereas other state of the art approaches requires a dual-arm manipulator. To perform the ironing task, we propose a robotic ironing method based on a full-body humanoid robot platform, using unmodified human tools in an unmodified domestic environment. Our method is focused on feasibility for a practical implementation in a real world domestic setting. For this purpose, the system relies on 3D perception and force/torque feedback. Garment data is obtained in a similar way to the previous approach, but including color data in the 3D reconstruction to perform garment segmentation. Once the garment has been segmented, a custom WiLD descriptor is computed that describes the wrinkleness of the garment surface to find the most wrinkled regions. Once these regions have been obtained, the most suitable ironing path is computed and executed with the robot, using force/torque feedback to have a precise control over the pressure exerted on the garment by the robot during ironing.

Acknowledgment

This work was supported by RoboCity2030-III-CM project (S2013/MIT-2748), funded by Programas de Actividades I+D in Comunidad de Madrid and EU and by a FPU grant funded by Ministerio de Educacin, Cultura y Deporte. It was also supported by the anonymous donor of a red hoodie used in our initial trials. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the NVIDIA Titan X GPU used for this research.

References

[1] R. Krantz-Kent, Measuring time spent in unpaid household work: results from the american time use survey, Monthly Lab. Rev. 132 (2009) 46.

[2] O. Khatib, Mobile manipulation: The robotic assistant, Robotics and Autonomous Systems 26 (2) (1999) 175–183. doi:10.1016/S0921-8890(98)00067-0.

[3] Y. Kita, F. Saito, N. Kita, A deformable model driven visual method for handling clothes, in: International Conference Pattern Recognition (ICPR), Vol. 3, IEEE, 2004, pp. 243–247. doi:10.1109/ROBOT.2004.1308874.

[4] D. Estevez, J. G. Victores, S. Morante, C. Balaguer, Towards Robotic Garment Folding: A Vision Approach for Fold Detection, in: IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2016.

[5] D. Estevez, R. Fernandez-Fernandez, J. G. Victores, C. Balaguer, Improving and Evaluating Robotic Garment Unfolding: A Garment-Agnostic Approach, in: IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2017.

[6] D. Estevez, R. Fernandez-Fernandez, J. G. Victores, C. Balaguer, Robotic Ironing With a Humanoid Robot Using Human Tools, in: IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2017.

[7] P. Jiménez, Grasp point localization, classification and state recognition in robotic manipulation of cloth: An overview, Robotics and Autonomous Systems 92 (2017) 107–125. doi:10.1016/j.robot.2017.03.009.

[8] A. Ramisa, G. Alenya, F. Moreno-Noguer, C. Torras, FINDDD: A fast 3D descriptor to characterize textiles for robot manipulation, in: IEEE International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 824–830. doi:10.1109/IROS.2013.6696446.

[9] F. Osawa, H. Seki, Y. Kamiya, Unfolding of Massive Laundry and Classification Types, Journal of Advanced Computational Intelligence and Intelligent Informatics (2006) 457–463.

[10] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, P. Abbeel, Bringing clothing into desired configurations with limited perception, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 3893–3900. doi:10.1109/ICRA.2011.5980327.

[11] B. Willimon, I. Walker, S. Birchfield, A new approach to clothing classification using mid-level layers, Proceedings - IEEE

International Conference on Robotics and Automation (2013) 4271–4278doi:10.1109/ICRA.2013.6631181.

[12] Y. Li, D. Xu, Y. Yue, Y. Wang, S.-f. Chang, E. Grinspun, P. K. Allen, Regrasping and Unfolding of Garments Using Predictive Thin Shell Modeling, in: International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 1382–1388.

[13] D. Triantafyllou, I. Mariolis, A. Kargakos, S. Malassiotis, N. Aspragathos, A geometric approach to robotic unfolding of garments, Robotics and Autonomous Systems 75 (2016) 233–243 doi:10.1016/j.robot.2015.09.025.

[14] Y. Kita, T. Ueshiba, E. S. Neo, N. Kita, Clothes state recognition using 3d observed data, in: International Conference on Robotics and Automation (ICRA), 2009, pp. 1220–1225. doi:10.1109/ROBOT.2009.5152741.

[15] Y. Li, Y. Wang, M. Case, S.-f. Chang, P. K. Allen, Real-time Pose Estimation of Deformable Objects Using a Volumetric Approach, in: International Conference on Intelligent Robots and Systems (IROS), IEEE, 2014, pp. 1046–1052. doi:10.1109/IROS.2014.6942687.

[16] A. Doumanoglou, T.-k. Kim, X. Zhao, S. Malassiotis, Active Random Forests: An Application to Autonomous Unfolding of Clothes, in: European Conference on Computer Vision (ECCV), Springer International Publishing, 2014, pp. 644–658. doi:10.1007/978-3-319-10602-1_42.

[17] J. Stria, D. Pruša, V. Hlaváč, Polygonal Models for Clothing, in: Advances in Autonomous Robotics Systems, Vol. 8717, 2014, pp. 173–184. doi:10.1007/978-3-319-10401-0_16.

[18] L. Sun, G. Aragon-Camarasa, S. Rogers, J. P. Siebert, Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening, in: IEEE International Conference on Robotics and Automation (ICRA), Vol. 2015-June, IEEE, 2015, pp. 185–192. doi:10.1109/ICRA.2015.7138998.

[19] L. Twardon, H. Ritter, Interaction skills for a coat-check robot: identifying and handling the boundary components of clothes, in: International Conference on Robotics and Automation (ICRA), 2015, pp. 3682–3688.

[20] J. Dai, P. Taylor, H. Liu, H. Lin, Folding algorithms and mechanisms synthesis for robotic ironing, International Journal of Clothing Science and Technology 16 (1/2) (2004) 204–214 doi:10.1108/09556220410520487.

[21] J. S. Dai, P. M. Taylor, P. Sanguanpiyapan, H. Lin, Trajectory and orientation analysis of the ironing process for robotic automation, Intl Journal of Clothing Science and Technology 16 (1-2) (2004) 215–226. doi:10.1108/09556220410520496.

[22] P. Kormushev, S. Calinon, D. G. Caldwell, Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input, Advanced Robotics 25 (5) (2011) 581–603. doi:10.1163/016918611X558261.

[23] Y. Li, X. Hu, D. Xu, Y. Yue, E. Grinspun, P. Allen, Multi-Sensor Surface Analysis for Robotic Ironing, in: IEEE International Conference on Robotics and Automation (ICRA), Stockholm, 2016. arXiv:1602.04918.

[24] S. Miller, M. Fritz, T. Darrell, P. Abbeel, Parametrized shape models for clothing, in: International Conference on Robotics and Automation (ICRA), 2011, pp. 4861–4868. doi:10.1109/ICRA.2011.5980453.

[25] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon, KinectFusion: Real-time dense surface mapping and tracking, in: 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2011, pp. 127–136. doi:10.1109/ISMAR.2011.6092378.

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

[27] R. B. Rusu, Semantic 3d object maps for everyday manipulation in human living environments, Ph.D. thesis, Computer Science department, Technische Universitaet Muenchen, Germany (October 2009).

[28] D. H. Douglas, T. K. Peucker, 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) (1973) 112–122.

[29] H. Digabel, C. Lantuéjoul, Iterative algorithms, in: Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, Vol. 19, Stuttgart, West Germany: Riederer Verlag, 1978, p. 8.

[30] W. Straßer, Schnelle kurven-und flächendarstellung auf grafischen sichtgeräten, Ph.D. thesis (1974).

[31] Z.-C. Marton, D. Pangercic, N. Blodow, M. Beetz, Combined 2D-3D categorization and classification for multimodal perception systems, The International Journal of Robotics Research 30 (11) (2011) 1378–1402. doi:10.1177/0278364911415897.

[32] T. Y. Zhang, C. Y. Suen, A Fast Parallel Algorithm for Thinning Digital Patterns, Communications of the ACM 27 (3) (1986) 236–239. doi:10.1145/5666.5670.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to algorithms, MIT press, 2009, p. 606.

[34] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, Journal of the society for Industrial and Applied Mathematics 11 (2) (1963) 431–441.

[35] S. Martínez, C. A. Monje, A. Jardón, P. Pierro, C. Balaguer, D. Muñoz, Teo: Full-Size Humanoid Robot Design Powered By a Fuel Cell System, Cybernetics and Systems 43 (3) (2012) 163–180. doi:10.1080/01969722.2012.659977.

[36] R. B. Rusu, S. Cousins, 3d is here: Point cloud library (pcl), in: Robotics and automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 1–4.

[37] L. B. Bridgwater, C. A. Ihrke, M. A. Diftler, M. E. Abdallah, N. A. Radford, J. M. Rogers, S. Yayathi, R. S. Askew, D. M. Linn, The robonaut 2 hand designed to do work with tools, 2012 IEEE International Conference on Robotics and Automation (ICRA) (2012) 3425–3430.

[38] J. J. Gago, J. G. Victores, C. Balaguer, Sign language representation by teo humanoid robot: End-user interest, comprehension and satisfaction, Electronics 8 (1). doi:10.3390/electronics8010057.
URL http://www.mdpi.com/2079-9292/8/1/57