A 3D descriptor to detect task-oriented grasping points in clothing

Arnau Ramisa, Guillem Alenyà, Francesc Moreno-Noguer and Carme Torras

Institut de Robotica i Informatica Industrial (CSIC-UPC),
Llorens i Artigas 4-6, 08028 Barcelona, Spain
{*aramisa, galenya, fmoreno, torras*}*@iri.upc.edu*

Abstract

Manipulating textile objects with a robot is a challenging task, especially because the garment perception is difficult due to the endless configurations it can adopt, coupled with a large variety of colors and designs. Most current approaches follow a multiple re-grasp strategy, in which clothes are sequentially grasped from different points until one of them yields a recognizable configuration. In this work we propose a method that combines 3D and appearance information to directly select a suitable grasping point for the task at hand, which in our case consists of hanging a shirt or a polo shirt from a hook. Our method follows a coarse-to-fine approach in which, first, the collar of the garment is detected and, next, a grasping point on the lapel is chosen using a novel 3D descriptor.

In contrast to current 3D descriptors, ours can run in real time, even when it needs to be densely computed over the input image. Our central idea is to take advantage of the structured nature of range images that most depth sensors provide and, by exploiting integral imaging, achieve speed-ups of two orders of magnitude with respect to competing approaches, while maintaining performance. This makes it especially adequate for robotic applications as we thoroughly demonstrate in the experimental section.

Keywords: 3D descriptor, Recognition, Detection, Grasping, Manipulation, Robotics

1. Introduction

Robotics research is increasingly addressing the challenges encountered by robots in human environments such as homes or offices. In order to be useful assistants, these robots need to be capable of recognizing and interacting with countless objects. These challenges further multiply when the objects to recognize and manipulate are not rigid, as is the case for garments: the infinite shape-state space of a garment is coupled with a wide variety of colors and designs.

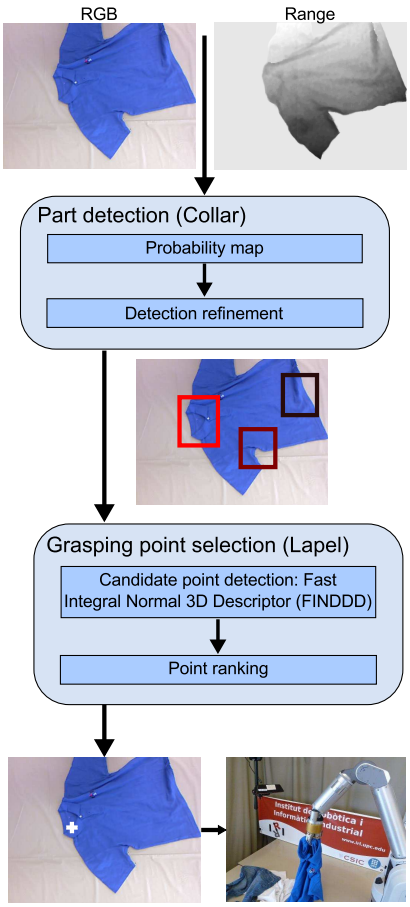


Figure 1: Schema of the proposed approach. Input RGB-D data is initially used to detect a certain garment part (the collar in this work) and then select a specific grasping point within the located garment part (on the lapel in this work).

Despite this inherent difficulty, in recent years the task of developing perception techniques that allow a robot to recognize and successfully manipulate textile objects has attracted attention from the research community. Yet, one drawback of most existing cloth manipulation methods is that they usually require a long series of re-grasps in order to bring the garment to the desired state, which can make the process slow. In contrast, we propose a new algorithm that uses state-of-the-art computer vision techniques to ensure a good initial grasp, so that a minimum number of re-grasps is required.

For demonstration purposes, in this work we consider the task of hanging a shirt from a hook with a single robotic arm equipped with a 2-finger gripper. In order to successfully grasp the garment and hang it from a hook, it is paramount to select a grasping point near the collar, preferably on the fold of the lapel in

the rear part, though any point in the lapel would also help simplify the task.

Because of the relatively small support region of the local descriptors, those using only appearance information suffer an important performance loss when generalizing to garments with color patterns not seen in the training data. On the contrary, descriptors based on 3D data are more robust and can more easily generalize to novel garments. Unfortunately, existing 3D descriptors [1, 2, 3] are not fast enough to accomplish this task in a reasonable time. To circumvent this problem, we propose the Fast Integral Normal 3D descriptor (FINDDD), a novel 3D descriptor that takes advantage of some constraint relaxations that can be made in a clothing manipulation scenario. In a series of experiments we show FINDDD to be two orders of magnitude faster to compute than state-of-the-art 3D descriptor while maintaining a similar performance. Furthermore, by using powerful and fast classification techniques, we can simultaneously accelerate the computation of our descriptor and push part of the computational effort to an offline classifier learning phase.

Another contribution of this paper is a complete perception system for selecting a point on the lapel of a shirt with high reliability, which, when grasped, will allow to hang it in a hook. Figure 1 shows the pipeline of the proposed method.

This paper is an extension of Ramisa et al. [4, 5] where a preliminary version of the part detection pipeline and the FINDDD descriptor were presented, respectively. The current paper combines and extends both previous contributions in a pipeline for garment manipulation, and it provides a more thorough mathematical description of the methods used, as well as a more extensive set of evaluations (e.g. influence of vocabulary size, PCA compression, larger datasets). Additionally, this work includes a grasping experiment that demonstrates the applicability of the whole approach in real settings.

2. Related Work

In this section we will briefly review existing work related, first, to garment manipulation and, next, to descriptors for 3D data.

2.1. Garment Manipulation

In recent years Maitin-Shepard et al. [6] and Cusumano-towner et al. [7] demonstrated functional end-to-end systems for automatically handling clothes (albeit in very controlled settings that simplify perception). The proposed systems are able to pick up a piece of laundry and manipulate it until a desired configuration is reached. Other approaches are more focused on the perception capabilities: Miller et al. [8] proposed a method based on parametrized shape models for estimating the pose of a crudely spread cloth item; in [9] the system is extended and combined with manipulation tools to complete the task of folding a garment with an open-loop sequence of movements.

Hidayati et al. [10] and Yamazaki et al. [11] proposed appearance-based garment identification systems, oriented to Internet shopping and to domestic

robots, respectively. Willimon et al. [12] exploited the manipulation capabilities of a robot to help in a perception task. Specifically, this work proposed a system to pick up the topmost element of a pile of clothing, and subsequently classifying it using interactive perception and four basic visual features. Later, Willimon et al. [13] developed a method to classify garments into six categories using a variety of appearance, depth, and mid-level features like "*round neck*" or "*front zipper*". These mid-level features allowed to significantly improve the classification performance; however, they cannot be automatically computed for a novel test image. Finally, very recently Doumanoglou et al. [14], proposed a system for unfolding certain items of clothing that used Random Forests and Hough Forests to determine a grasping point from a canonical view (i.e. the "absolute lowest point" of a hanging garment) using a bimanual manipulator.

In contrast to the previous work, our proposed technique models characteristic parts of garments, aiming at identifying a suitable grasping point prior to any manipulation attempt, and generalizing beyond the clothes used for training.

2.2. 3D Descriptors

With the recent popularity of 3D cameras, a growing number of local 3D descriptors that could be used for robotic manipulation tasks have been proposed. However, they are mostly targeted to rigid objects and unstructured point clouds (i.e. sets of 3D points with no a priori known neighborhood relations), which makes them computationally expensive, especially when the descriptors need to be densely computed.

Traditionally, 3D descriptors were applied to synthetic CAD models [15, 16] and worked on unstructured point clouds. One of the earliest related approaches, also evaluated on synthetic range images, was proposed in [17], where a histogram representation of a complete point cloud was built merging information of the pixel depths, normal orientations and surface curvatures. More recently, Flint et al. [18] defined a descriptor for Hessian-based interest points by accumulating in a histogram the elevation difference of the normals estimated with different sized planes for all points in a support region. Flitton et al. [19, 20] extended the SIFT descriptor to 3D complex CT imagery and used it to perform object detection in airport baggage control using a Bag of Visual Words model.

Given an unstructured point cloud, the Normal Aligned Radial Feature (NARF) descriptor [21] first computes a normal aligned range image centered on an interest point, where points on a local neighborhood are projected onto a plane along the normal direction. The descriptor is then built according to the variation occurring among a number of rays projecting into these images. View-point independence is achieved by normalizing to a canonical orientation.

Following a similar philosophy as the SIFT descriptor for 2D images, Tombari et al. [3] presented the Signature of Histograms of OrienTations (SHOT). Given an interest point, its 3D neighborhood is split into a fixed number of regions, and the descriptor is built based on histograms of differences between the normals at the points within the region and the normal at the interest point.

Rusu et al. [22] introduced the Point Feature Histogram (PFH) descriptor. It is based on four angle relations computed between every pair of points in a

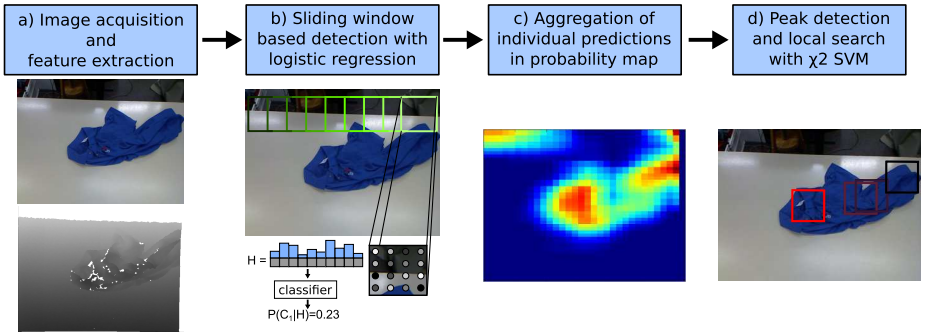


Figure 2: Schema of the part detection method (collar in this figure). Steps *b*-*d* correspond to the different layers of the approach as described in the text. In the image of step *d*, reddish color of the bounding box indicates more confidence in the detection.

k-neighborhood. Each relation is accumulated in a 16-dimensional histogram, yielding a descriptor which is shown to be invariant to position, orientation and point cloud density. Yet, the cost of computing n descriptors on a point cloud is $O(n \cdot k^2)$. Later, the same authors proposed the Fast Point Feature Histograms (FPFH) [1], which instead of computing the relation between every pair of points in a neighborhood, it only considers the connection between the point of interest and its neighbors, and re-weights the result with descriptor information from the surrounding points. This reduces the cost to $O(n \cdot k)$. Despite being faster, the cost to compute a single FPFH is still high for real-time applications, and it is not applicable to very dense point clouds or situations where one might want to compute descriptors covering a large area.

Zhang et al. [40] proposed the Histogram of 3D facets (H3DF) descriptor for hand gesture recognition. To compute this descriptor, first the point cloud of the hand is segmented from the whole scan and normalized according to the dominant orientation of the depth gradient. Next, 3D facets are computed as the normals of local regions, which are then coded by projecting the normals onto three orthogonal planes. Finally, the final H3DF descriptor is constructed by pooling all coded normals using a concentric schema that covers the entire hand point cloud.

Sun et al. [39] used a high-quality stereo system to acquire a dense depth map of a piece of clothing laying on a table. After fitting a B-Spline model to the depth data, they construct a descriptor for every wrinkle in the cloth that captures the width, height, length and volume, as well as the type of wrinkle, and use this information to plan a dual-arm flattening strategy.

Summarizing, most of the previous descriptors are designed to work on unstructured point clouds, thus a significant part of their computational cost is associated to defining the neighborhood in which the descriptor has to be computed and to establishing a reference frame to obtain invariance to viewpoint. The consequence is that most of them are just computed for a few points of interest. This is in contrast to the 3D descriptor we propose here, which exploits

the spatial neighborhood relations of the 3D sensor plane, leading to very fast computations even when the descriptors are densely computed.

3. Part Detection: Collar

This section describes the methodology to detect the collar of a shirt or polo shirt, which corresponds to the first step in the pipeline outlined in Fig. 1. The method follows a coarse-to-fine strategy, which starts by approximately locating the relevant area of the garment to then refine the initial detection with a more precise classifier. See the schematic in Fig. 2.

Concretely, in the first step of the part detection method (Figure 2b), the appearance and/or depth local descriptors of choice are extracted at regular positions over the image, and quantized into visual words using a vocabulary of K centroids, previously computed using k-Means in a large collection of training descriptors. The image is then described using "Bag of Visual Words" (BoVW) [26] vectors computed in a collection of sub-windows, organized in a grid over the image and filtered by a segmentation mask computed combining color and depth data. More formally, let $L = \{l_1, \ldots, l_N\}$ be a set of sub-windows of an RGB-D image $\mathcal{I}$. We can then define the BoVW histogram of sub-window l_j as:

$$\mathbf{h_j^*} = [\nu(v_i, l_j); i = 1, \ldots, K] \tag{1}$$

where ν is the *frequency* function that returns how many times a certain visual word v_i appears inside the sub-window l_j. Next, the histogram is normalized and square-rooted:

$$\mathbf{h_j} = \sqrt{\frac{\mathbf{h_j^*}}{||\mathbf{h_j^*}||_1}} \tag{2}$$

This operation, also referred to as **power normalization**, has been shown to better approximate more realistic non-independent and identically distributed models of visual words in natural images [27]. Multiple descriptors, for example computed using appearance and depth information, can be straightforwardly combined by simply concatenating the respective bag of visual words (BoVW) vectors $\mathbf{h_j}$.

Next, a logistic regression model with parameters $\mathbf{w}$ is learned by minimizing the following expression on a training dataset of image sub-windows:

$$\min_{\mathbf{w}} \left(\frac{1}{2}\mathbf{w^\intercal w} + C \sum_{i=1}^{T} log(1 + e^{-y_i \mathbf{w^\intercal h_i}}) \right) \tag{3}$$

where C is the regularization parameter (adjusted by cross-validation), $\mathbf{h_i}$ stands for the i_{th} BoVW vector coming from the set of T training vectors, and y_i is +1 or -1 indicating if $\mathbf{h_i}$ comes from a positive (i.e. contains the desired part) or negative sub-window. The learned logistic regressor is then used to obtain

the posterior probability of the part presence in each sub-window. The part presence probability (class C_+) for sub-window l_j is computed as:

$$p(C_+|\mathbf{h_j}) = \frac{1}{1 + e^{\mathbf{w}^\intercal \mathbf{h_j}}} \tag{4}$$

Probability map:. To reduce noise and aggregate local information, instead of directly selecting the highest scored windows, an initial set of locations Θ is selected as all the local maxima of the average per-pixel probability based on the overlapping sub-windows (Figure 2c):

$$\mathcal{M}(x) = \frac{\sum_{j=1}^{N} \delta(x \in l_j) P(C_+|\mathbf{h_j})}{\sum_{j=1}^{N} \delta(x \in l_j)}; \forall x \in \mathcal{I} \tag{5}$$

$$\Theta = \{x \mid \mathcal{M}(x) > \mathcal{M}(\mathcal{N}(x, \mathcal{I})); \forall x \in \mathcal{I}\} \tag{6}$$

where x is a pixel position in the input RGB-D image $\mathcal{I}$, $\delta(.)$ is the indicator function and $\mathcal{N}(x, \mathcal{I})$ is the 8-neighborhood of a point x in $\mathcal{I}$.

Detection refinement:. In the third stage of the approach (step d of Figure 2), to overcome the limitations of linear logistic regression, the locations found in the previous stage are refined via a Support Vector Machine using a RBF kernel with χ^2 distance:

$$\chi^2(\mathbf{q}, \mathbf{t}) = exp(-\gamma \sum_j \frac{(q_j - t_j)^2}{q_j + t_j}) \tag{7}$$

where γ is the inverse of the average of the χ^2 distance between the vectors of the training set, and t_j and q_j are the j_{th} components of two BoVW vectors. A set of sub-windows are cast with different areas, aspect ratios and offsets with respect to the original points in Θ, and their corresponding BoVW vectors are directly evaluated using the non-linear classifier. The highest scored window θ for each peak is selected and the corresponding posterior probability is approximated from the classifier score (z_θ) with a sigmoid function:

$$P(C_+|\mathbf{h}_\theta) \approx \frac{1}{1 + e^{Az_\theta + B}} \tag{8}$$

where A and B are the sigmoid parameters, adjusted using Platt's probabilistic output algorithm [28, 29].

4. Grasping Point Selection: Lapel

Depending on the task to be conducted after grasping the garment, mindfully selecting the grasping point can have a significant impact on difficulty; hence, rather than a generic method, we use a specific strategy tailored to find a suitable grasping point for the task at hand (see Fig. 1).

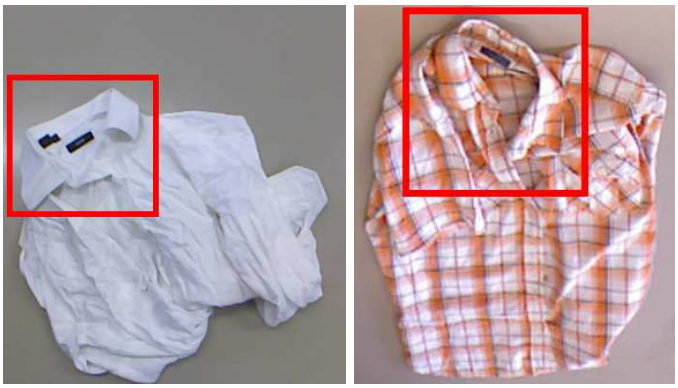


Figure 3: Garments with very different appearance may reduce the performance of intensity based descriptors such as SIFT. In this figure we can see two very differently textured shirts with the ground truth bounding box around the collar.

Candidate point detection. In our case the objective is to hang the garment from a hook, which becomes much easier if the manipulator has the shirt grasped by the lapel of the collar. We consider all points inside of the detected collar area (using the method presented in the previous section) as candidate points, and a local descriptor is computed for each one in order to determine the final grasping point.

Point ranking. In order to detect the final grasping point, we train a logistic regressor able to distinguish between lapel and non-lapel, and use it to rank the candidate points. The one that maximizes the regression score (as in Eq. 4) is selected as final grasping point. However, using appearance information for this task can lead to poor results if the texture of the garments used for training data is significantly different from the ones used for test. A solution to this problem is to use primarily depth data to detect the final grasping point. As an example, in Table 1 the performance of depth (SHOT) and appearance (SIFT) descriptors exemplifies how training the classifier using plain shirts and testing on a shirt with texture degrades the performance of the appearance-based classifier without affecting the depth-based one. The set used in the experiment included 125 images for training and 100 for testing (50 textured and 50 plain), all drawn from the dataset used in the remaining experiments [1]. Hence, enriching models with depth information should yield improved results in garment perception tasks.

Unfortunately, existing 3D descriptors are too slow to be densely extracted in a reasonable time, and a fast perception-action cycle is crucial in robotics applications. To address this problem we present a novel and fast 3D descriptor for clothing manipulation, which is described in the next section.

[1] Available at `www.iri.upc.edu/groups/perception/#clothingDataset`

Table 1: Accuracy (%) of lapel point detection when training using plain white and gray shirts and testing on plain and textured shirts (see Figure 3).

	Descriptor	Plain	Textured
Lapel	SHOT	72	68
(Accuracy)	SIFT	74	42

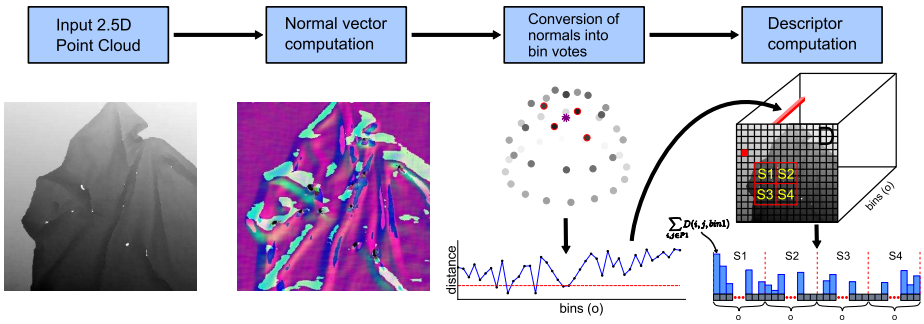
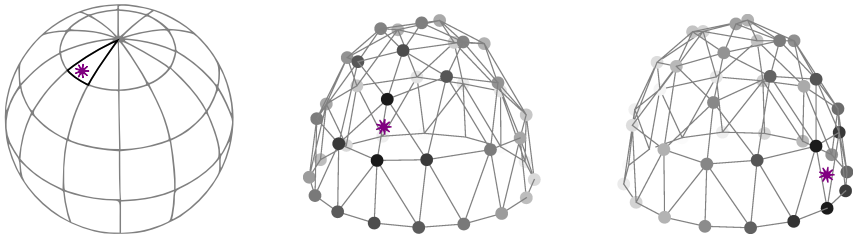


Figure 4: Steps of the FINDDD descriptor computation. The second image shows the normal vectors mapped into RGB. The third image shows the orientation bin centers colored according to distance to the input normal, plotted as a purple asterisk. The red line in this plot is the threshold above which the contribution to bin centers is 0 (corresponding bin centers highlighted in red). See the text for more details.

5. Fast Integral Normal 3D Descriptor

We propose the Fast Integral Normal 3D (FINDDD) descriptor, that takes advantage of the specificities of a clothes manipulation scenario to be both highly discriminative and very fast to compute. First, since the range of depth values within the object will be very limited, we can safely assume that the density of points will be approximately constant over the whole cloud (i.e. the separation between neighboring points in the image plane depends on their respective depths. Thus, similar depths will produce similar inter-point distances, and hence similar point densities). Next, given that all points in a structured point cloud are distributed in an equally-spaced grid or image-like organization, adjacency is well defined. Finally, given the design of our approach and in contrast with other 3D descriptors such as SHOT, we do not need a specific coordinate frame for each descriptor, typically used to ensure invariance to rotations. Instead, we rely on the capacity of the learning model used to classify the descriptors and subsume the variations induced by changes in viewpoint. Furthermore, additional synthetic training data could be easily generated from existing 3D scans simply by applying rotation transformations, in a similar way as done (in a 2D context) for appearance descriptors [30, 31, 32].

The combination of the aforementioned assumptions allow for very fast computations using integral images. We believe that for a scenario such as robotic manipulation of textile objects, a rapid perception cycle may be more relevant



(a) (b)

Figure 5: Bin distributions resulting from (a) angular thresholds in spherical angles (bins represented as areas) and (b) the adopted tessellation-based approach (bin centers represented as dots). Our 3D descriptor uses a parametrized division of the sphere to create the voting bins that avoids the typical meridian division that over-represents the zone near the north pole. Examples of vote patterns are shown, where the purple asterisk indicates the coordinates of the normal vector.

than highly discriminative but very expensive descriptors. Figure 4 summarizes the steps in the computation of the descriptor, that will be explained in more detail in the following paragraphs.

Taking inspiration from the SIFT descriptor [33], we define FINDDD as the concatenation of normal vector (i.e. surface) orientation histograms for several sub-regions inside a support area around the point of interest. The steps to compute the descriptor can be summarized as follows:

1. The normal vector $\mathbf{n_{xyz}}$ of every point in the input structured point cloud is computed. This step can be made very fast using integral images.
2. For each normal vector computed in the previous step, we construct a vector $\mathbf{u}$ with components based on the distance between $\mathbf{n_{xyz}}$ and a set of o orientation bin centers.
3. To compute a descriptor with support region $\mathcal{S}$, divided in s sub-regions $\mathcal{S} = \{S_1, \ldots, S_s\}$, all $\mathbf{u}$ within a sub-region are added into a vector $\mathbf{F_i}$.

$$\mathbf{F_i} = \sum_{\mathbf{u} \in S_i} \mathbf{u} \tag{9}$$

   Then, the full descriptor is formed by concatenating all sub-vectors: $\mathcal{F} = [\mathbf{F_1}, \ldots, \mathbf{F_s}]$.
4. Finally, each descriptor is normalized using the L1 norm to make it robust to different densities in the number of points ("NaN" points due to missing depth data, caused by occlusions or noise, are discarded). Like in the case of SHOT [3], we found it beneficial to keep local density information by only normalizing at the global level.

5.1. Orientation Assignment

In order to re-use previously computed data, and in contrast with other works that also use point normal information [3], we do not accumulate the angle between the normal at every point and the normal at the central point of

the descriptor support area. Moreover, since we are dealing with 2.5D data, only half of the sphere of orientations has to be considered, which further reduces the size and computational cost of the descriptor.

Since the normal vectors are circumscribed to the unit sphere, a common strategy is to express them as angles in spherical coordinates (e.g. [17]):

$$(\phi, \theta) = \left(arccos\left(\frac{n_z}{r}\right), arctan\left(\frac{n_y}{n_x}\right)\right) \tag{10}$$

$$r = \sqrt{n_x^2 + n_y^2 + n_z^2}. \tag{11}$$

where ϕ is the inclination and θ is the azimuth, and (n_x, n_y, n_z) are the normal vector coordinates.

However, defining the orientation bins in the angular space has some caveats: first, bins do not cover the same sphere space in all locations (see Figure 5a), and are smaller around the north pole (maximum elevation), which leads to a non-uniform representation of the normals; second, azimuth information becomes unstable as vectors get closer to the maximum elevation point, and small changes due to noise can easily produce swaps in the assigned bin.

Instead, we define bins distributed across the entire semi-sphere in Cartesian coordinates. Precisely, we use the vertex points generated in a triangular tessellation to obtain a quasi-regular distribution of the orientation bins (see Figure 5b). One alternative yielding completely regular bins is the approach of Klaser et al. [34], where points in the sphere surface are projected onto a platonic solid; however, it has a limitation on the number of bins, since the platonic solid with more facets available is the icosahedron (20-sided).

The downside of our representation is a higher cost to assign a normal to its corresponding bin, because the distance to the bin centers in the unit half-sphere surface has to be computed. However, for a reasonably small number of bins, there is no noticeable slow-down in computation and, if a larger number of bins is desired (e.g. for data coming from a very precise 3D sensor), structures like K-D Trees can significantly accelerate the search, or computational geometry techniques could be used to directly compute the triangle into which the normal vector projects.

Increasing the number of orientation bins of the descriptor improves the angle resolution of the model, and consequently the accuracy with which surfaces can be represented. However it also induces aliasing and sparsity, which degrade the significance of distances between descriptors. Another consideration regarding the number of bins is that it must be adjusted to the level of noise inherent in the input data, which may otherwise worsen the aliasing problem.

We mitigate these two problems using soft voting to interpolate between different bins [33]. Each normal contributes to all bins closer than a unit of bin spacing:

$$\mathbf{u} = \left[\max\left(1 - \frac{||\mathbf{b_i} - \mathbf{n_{xyz}}||}{\lambda},\ 0\right); i = 1, \ldots, o \right], \tag{12}$$

where $\mathbf{u}$ collects the votes for each centroid $\mathbf{b_i}$, $\mathbf{n_{xyz}}$ is the normal vector and λ is the distance between neighboring bin centers.

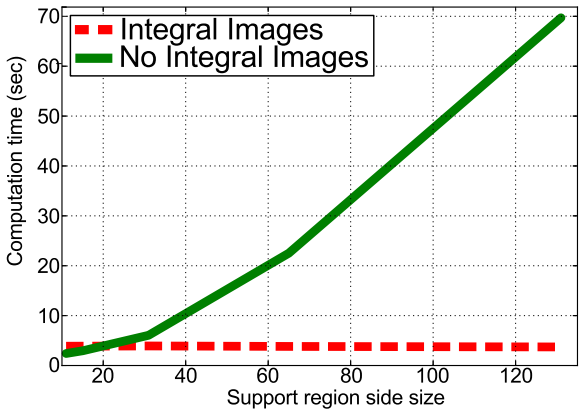


Figure 6: Comparison of computation time for descriptors extracted densely (every pixel), varying the size of the local region used.

5.2. Efficient Computation Using Integral Images

As mentioned earlier, by using structured point clouds, it is possible to take advantage of integral images[2], and make the computational cost linear in the final number of descriptors n in practice, involving only $O(o \cdot n \cdot s)$ operations, where s and o are typically negligible: the former is the number of spatial sub-divisions (typically 16, for a 4×4 grid), and the latter is the number of orientation bins (13 in most of our experiments). We use integral images both to compute the normal vectors[3] and to perform the aggregation of the votes for every orientation bin and sub-regions of the descriptor.

To compute the descriptors, one integral image is necessary for each orientation bin, where we will accumulate the occurrences of normals falling into it from the top-left to the bottom-right corner of the structured point cloud. Consequently, the cost of constructing the integral images is $O(o \cdot p)$, where p is the total number of points in the point cloud.

Figure 6 shows a comparison of the computational time required to extract a descriptor for every point of a 640×480 structured point cloud, with and without integral images, using a 3Ghz Linux machine. Note that in the case where integral images are not used, the neighborhood information provided by the structured point cloud is still used and, in an unstructured point cloud, we would require an extra step of searching for the nearest neighbors of each point in which we want to compute a descriptor. This would require typically using a K-D tree for nearest neighbor search, at an additional cost of $O(log(p))$ for

[2]This is the name by which the technique of summed-area tables became popular in the computer vision community, and is used here for this reason. This technique is in fact applicable to any matrix-like object (structured point clouds in our case).

[3]To compute the normal vectors using integral images, we use the implementation in the PCL 1.5 library, based on [23, 24, 25].

each descriptor, plus $O(p \cdot log(p))$ for building the tree once.

5.3. Compression using PCA

Using 13 orientation bins and 4×4 spatial subdivisions, the descriptor has a total of 208 components. This size may not seem very large, since other well known descriptors have a similar size (e.g. SIFT and SURF have size 128, GIST has size 960 and SHOT has size 352). On the other hand, FPFH has only 33 components and yields state-of-the-art results in our tests. Furthermore, it has been shown by Sanchez et al. [35] that using Principal Component Analysis (PCA) to compress SIFT descriptors to 64 dimensions actually boosts the results for image classification tasks, since it helps de-correlate the descriptors and reduce the effect of noise.

Looking at the eigenvalues computed with a large dataset of FINDDD descriptors, we see that the energy quickly goes down, and is almost zero at the fiftieth eigenvalue. Consequently, we also evaluated the effects of projecting the FINDDD descriptors using PCA to the same dimensionality as FPFH, that is, 33 components, and found that the performance of the descriptor not only remains stable, but it even improves in some cases.

6. Experimental Results

We have evaluated the proposed method in a series of experiments. First we evaluate and compare the FINDDD descriptor with other state-of-the-art 3D descriptors, and subsequently we individually evaluate the different parts of the approach: the collar detector and the FINDDD descriptor for lapel point selection. Finally, the complete system is integrated in a robot manipulator and evaluated in a cloth grasping experiment.

6.1. FINDDD Benchmarking

We evaluated the performance of FINDDD compared to other state-of-the-art 3D descriptors, and found it to have a comparable behavior in terms of accuracy in several tasks related to garment perception, while being two orders of magnitude faster than the competing descriptors.

6.1.1. Efficiency Assessment

To evaluate the computational cost of our descriptor, we measured the time necessary to extract a descriptor for every point of a 640×480 structured point cloud acquired with a Kinect range camera on a 3Ghz Linux machine (average of the 10 first point clouds of the dataset from [36], the time reported also includes loading the point cloud, computing the normal for every point and storing the results back to disk). Points at the edge of the point cloud, without enough neighbors for the support region, were discarded. Using more orientation bins increases the number of integral images that have to be constructed and the operations to compute each sub-region. Consequently, we have tested the method with 13 and 41 orientation bins.

Table 2: Comparison of the computational cost of the evaluated descriptors. Parameter o for FINDDD stands for the number of orientation bins considered. Refer to the main text for more details.

Descriptor	Time (s)	Time per desc. (ms)
FINDDD (o=13)	4.4 ± 0.23	0.0168 ± 0.00086
FINDDD (o=41)	11.0 ± 0.41	0.0441 ± 0.00157
FPFH	352.1 ± 75.0	1.44 ± 0.047
SHOT	581.6 ± 14.9	2.37 ± 0.294

Two other state-of-the-art descriptors, SHOT and FPFH[4], are also evaluated to provide a reference to our results. The setup is the same as the one used for our descriptor, but in this case points with a NaN value in any coordinate were filtered out from the input point cloud as advised in the documentation of the descriptors.

Table 2 shows the results of the comparison. In the table, the column **Time** shows the time spent computing the descriptors for the whole point cloud, and column **Time per desc** the average time for a single descriptor (an upper bound, since the disk reading/writing overhead is also included). Note that FINDDD, in its both configurations, is near to two orders of magnitude faster than both FPFH and SHOT. We will next show that the recognition rates when characterizing garments, are very similar for all methods.

6.1.2. Wrinkle Identification

We perform an initial evaluation of the proposed descriptor performance on an in-house dataset containing 640×480 Kinect RGB-D images of a polo shirt showing one of eight distinct manually produced wrinkles [5]. Five repetitions of each wrinkle were acquired, and the relevant wrinkle area in each image was annotated by hand. Then, we extracted pixel-wise descriptors for each image, and selected the center of gravity of the annotation as the representative for the particular wrinkle and image. We also stored a fixed number of descriptors from random points in the annotated regions for additional testing.

Using this dataset we evaluate the retrieval performance of FINDDD. The distance to a query descriptor is used to re-order all the descriptors in the dataset, and the average precision (AP) of the resulting list is computed. The AP is then averaged over every representative instance of every wrinkle type. The same queries are performed in two datasets: one where only the representative descriptors are present (columns *Rep* of Table 3), and another that additionally contains the descriptors of all the previously selected random points (columns *Ext* of Table 3). We also evaluated the performance of the FPFH and SHOT descriptors in the same way, varying the radius of the region used to

[4]Both descriptors were computed using the PCL 1.5 Library, and with the parameters suggested in the tutorials of the library (i.e. support region of 5cm).

[5]Available at `http://www.iri.upc.edu/groups/perception/#finddDescriptor`

Table 3: Results of a retrieval experiment. *o* stands for the number of orientation bins used, *s* for the side (in pixels) of the support region, and *sv* shows if soft voting was used or not (True/False). Column *Rep* shows the mean average precision (%) of the test where only the representative points were used, while column *Ext* shows the results for the test that included the additional random points.

			FINDDD			FPFH			SHOT	
o	*s*	*sv*	*Rep*	*Ext*	*Rad*	*Rep*	*Ext*	*Rad*	*Rep*	*Ext*
13	21	T	52.6	50.1	3cm	46.1	50.8	3cm	27.0	25.3
13	21	F	45.7	45.8	4cm	45.3	53.7	4cm	34.8	30.8
13	43	T	62.8	69.8	5cm	44.8	53.6	5cm	35.7	33.9
13	43	F	56.1	64.6	6cm	42.2	52.4	6cm	34.5	34.7
13	65	T	**67.6**	76.1	7cm	39.0	50.7	7cm	36.2	35.1
13	65	F	60.3	71.6	8cm	36.2	49.2	8cm	34.1	35.3
41	21	T	52.7	52.2	9cm	34.8	47.7	9cm	33.8	36.1
41	21	F	48.1	49.0	10cm	33.1	46.8	10cm	33.7	36.5
41	43	T	66.5	74.6	15cm	33.4	45.1	15cm	30.9	37.7
41	43	F	62.6	70.9	20cm	34.2	44.8	20cm	27.2	36.0
41	65	T	67.0	**77.8**	25cm	32.0	43.6	25cm	25.5	34.3
41	65	F	63.8	75.4	30cm	30.9	42.7	30cm	23.4	32.0

compute the descriptor. Different tasks and descriptors exhibit different optimal region sizes, with larger regions generally yielding better results in the trials with the additional sampled points (Ext).

From the results in Table 3 it can be seen that, in our textile manipulation setup, our proposed descriptor is able to correctly characterize and recognize wrinkles with a performance similar or even superior to that of the state-of-the-art descriptors. It is also noticeable that being able to use a large enough support region is essential to properly characterize a textile wrinkle, which our descriptor is able to do at no additional computational cost.

6.1.3. Wrinkle Segmentation

In this section we show that the FINDDD descriptor is able to accurately characterize the different types of wrinkles present in a garment. Figure 7 shows a visualization where pixels in a range image of a t-shirt are color-coded according to the label assigned to their corresponding FINDDD descriptor by spectral clustering, with different numbers of clusters. As can be seen, the information encoded in the FINDDD descriptor can be useful to clearly identify the main 3D structures, such as ridges or saddles, with increasing levels of detail. However, this is still preliminary work, and some subtleties may fail to be properly captured by the labeling.

This could then be potentially used to model the global state of the object and for planning tasks such as flattening the garment.

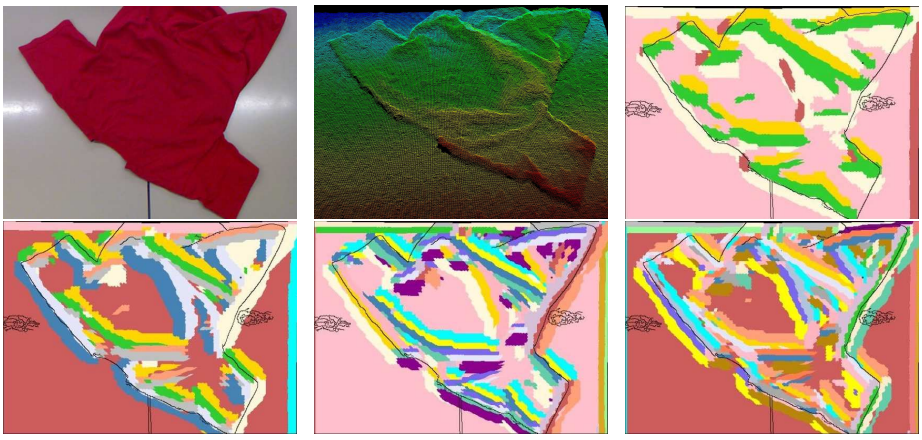


Figure 7: Pixel-wise labellings of a t-shirt range image showing the result of spectral clustering applied on the FINDDD descriptors. Original image and point cloud shown in the top-left position and number of clusters is, from left to right and top to bottom: 5, 10, 20, 25. Colors are assigned randomly in each image, and Canny edges are superimposed for clarity. Best viewed in color.

6.1.4. Garment Recognition

We also compared the proposed descriptor with FPFH and SHOT in a class recognition task consisting in distinguishing clothes made of different types of textile materials on the basis of the wrinkles they produce. We used the publicly available *IRI Clothing Part* dataset [36], that contains over a thousand RGB-D scenes of six classes of garments (polo shirt, jeans, t-shirt, hooded sweater, shirt and dress), with precise segmentation mask annotations for the collar and other parts of the garments. The clothes are lying on top of a table, and are displayed in different degrees of wrinkledness.

We split the dataset in two parts (70% train and 30% test), and represented the images using BoVW models of size 512. Then we trained a Support Vector

Table 4: Average precision (%) obtained by the different 3D descriptors in the garment recognition task.

	Linear SVM			RBF-χ^2 SVM		
Garment	FINDDD	SHOT	FPFH	FINDDD	SHOT	FPFH
Dress	37.5	25.5	**51.2**	66.8	61.9	**67.6**
Shirt	41.7	41.2	**45.1**	54.5	72.9	**79.7**
T-Shirt	**71.8**	58.1	68.9	**84.7**	70.1	76.5
Jeans	41.5	33.4	**58.1**	72.9	65.1	**77.9**
Polo	**85.2**	64.7	71.6	**96.0**	83.7	77.6
Sweater	44.8	36.5	**80.1**	84.6	92.1	**93.7**
Average	53.7	43.2	**62.5**	76.6	74.3	**78.8**

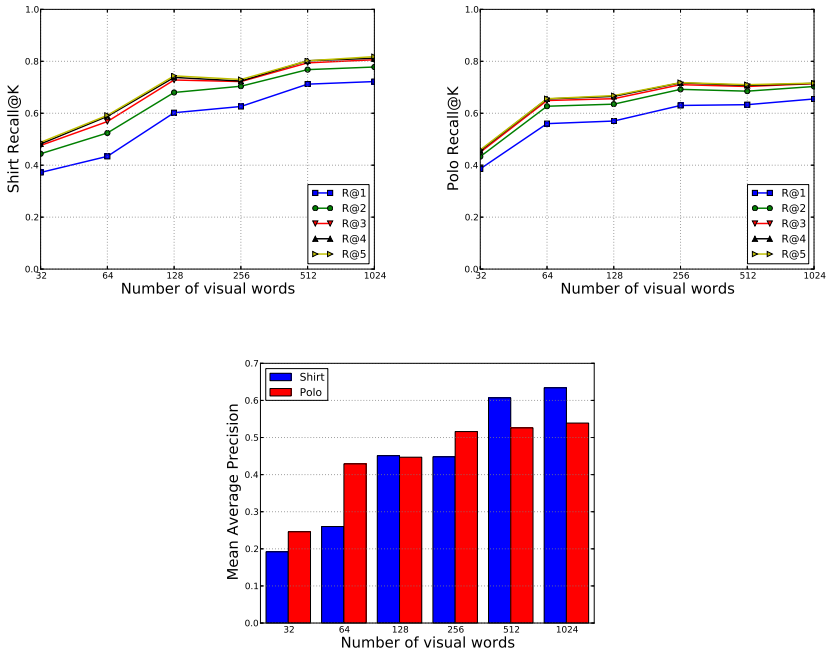


Figure 8: Collar detection results using visual vocabularies of increasing sizes. The top two figures show the recall@1-5 (shirt and polo shirt, respectively), and the bottom one the mean Average Precision for both classes.

Machine (SVM) classifier with either linear or RBF-χ^2 kernels on the BoVW for each textile material class. The average precision obtained by FINDDD (13 orientations and 43-pixel sided support regions), FPFH and SHOT descriptors in this test can be seen in Table 4. As can be observed, the performance of FINDDD is comparable to that of the two state-of-the-art descriptors. The superior performance with the polo shirt may be attributable to the higher number of training samples for that category.

Note that this test is not directly comparable to the recent work by Willimon et al. [13], since they are doing clothing classification with a dataset focused on intra-class variation using a sophisticated approach containing both appearance and depth information, as well as mid-level feature information, while here we distinguish between different exemplar garments using only depth information.

6.2. Collar Detection

As explained in Section 3, we use the part detection method to have an initial estimate in which we can restrict the search for the fine-grained grasping point location. For simplicity, and given that directly incorporating depth descriptors does not significantly improve the test accuracy of this operation [37], we use only SIFT descriptors to construct the BoVW representations. In the future,

Table 5: Accuracy (%) of lapel point selection when using different spatial priors. In the *Ground Truth (GT) box* the annotated bounding box is used to restrict the search, while in the *Detect* column the bounding box obtained in the collar detection step with SIFT is used. We also tested the performance without using a spatial prior with FINDDD 33 (for computational reasons), that can be seen in the *Full ima* column.

	Shirt			Polo		
Descriptor	Full ima	GT box	Detect	Full ima	GT box	Detect
FINDDD	-	74.4	80.3	-	63.1	61.2
FINDDD 33	18.4	74.4	82.2	13.9	67.2	59.5
FPFH	-	40.6	40.0	-	45.5	48.8
SHOT	-	72.2	66.3	-	62.7	54.1
SIFT	-	74.4	77.5	-	68.3	68.9

we plan to combine shape and appearance information in a way that exploits the advantages of each source of information [38].

We evaluate the performance of this detection method using the relevant part of the *IRI Clothing Part* dataset [36] (i.e. scenes that show shirts or polo shirts).

Results can be seen in Figure 8. An experiment is considered a true positive if the predicted grasping point falls within the annotated ground truth bounding box, and the performance is measured using two metrics: Recall at K ($R@K$), which tells for which percentage of the images a true positive is found on the K top scored detections. This measure, and specially $R@1$, is of most relevance in flexible object manipulation, where typically we will only have one chance of correctly grasping the object before altering its state, hence requiring a new detection process. The second metric used is mean Average Precision, which is equivalent to the area below the precision-recall curve[6], and takes into account both precision and recall. As can be seen, the main elbow in performance of the collar detection system is close to 128 visual words for both the shirt and the polo shirt, where around 60% recall is attained. The good performance attained with this relatively low number of visual words, at least compared to typical values used in visual object classification literature (e.g. see the contestants in any edition of the Pascal Visual Object Challenge), can be explained by the limited "visual world" displayed in the images. Most failure cases of the method are examples where the garment was highly wrinkled, or where the collar is only partially visible due to folding of the clothes. Images in the test set that included garments not used in training also had a higher failure rate, specially those with long sleeves, which sometimes fold in ways that can mimic the collar appearance. See Figure 9.

[6] An evaluation metric that shows the precision attained at every level of recall, when ordering the test samples by their predicted score.

Figure 9: Examples of images where the collar detection method may typically fail.

6.3. Lapel Point Selection Using FINDDD

The last step of our approach (see Figure 1) consists on exploiting 3D data to select a final grasping point in the lapel of a shirt or polo shirt within the box enclosing the detected collar. This selection can be done based solely on the "graspability" of the point, but it is also possible to use other criteria that facilitate the subsequent task. For instance, hanging the garment from a hook is greatly simplified if it is grasped by a point on the lapel of the collar.

To evaluate how well our proposed descriptor can detect this type of fine-tuned grasping point compared to other 3D descriptors, as explained in Section 4, we trained a logistic regression linear classifier using manually annotated lapel and non-lapel points within the designated collar area. Then, we computed their accuracy on a separate testing set. All points within the collar area are evaluated, and the one with highest score is predicted as final grasping point.

As can be seen in Table 5, the FINDDD descriptor (13 orientations and 43-pixel sided support regions) obtains very competitive results, outperforming FPFH and SHOT at this task, even when projected to 33 dimensional subspace using PCA, which successfully removes the noise and unused dimensions of the descriptor, and in most cases maintains or improves the performance. Results using SIFT appearance descriptors are also reported for comparison purposes.

Two spatial priors, that provide different insights, are used. On the one hand, the *Ground Truth (GT) box* reflects the case of an optimal collar detector: it is always accurate, but usually larger, and is available for all images. On the other hand, the *detected box* (detection is done using SIFT and a 128 visual word vocabulary) shows the performance of the lapel classifier in the actual system: the evaluated box is noisier and may not cover the whole collar area, but is smaller and, since the lapel is only detected when the collar detection step is successful, it leaves out the hardest images.

The benefit of a two-stage approach (i.e. first detecting the collar and then selecting the fine-grained grasping point) is apparent when compared to applying the lapel point classifier to the whole image. In that case, the percentage of correct grasping point detections is reduced to 13.9% for polo shirt, and to 18.4% for shirt.

The random chance baseline for this task would be 35.8% accuracy, computed by looking at the relative area of annotated pixels inside the ground truth bounding boxes.

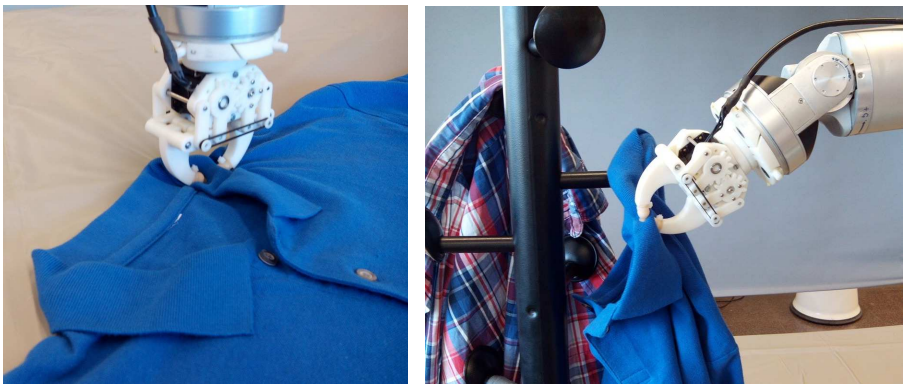


Figure 10: Detail of the gripper and the hanging operation.

Table 6: Detection results of the lapel grasping experiments using different numbers of visual words in the collar detection step. The *Lapel* column shows the accuracy (%) of selecting a grasping point on the lapel considering the cases in which the collar was correctly detected (*Collar* column).

	128 vw		1024 vw	
Garment	Collar	Lapel	Collar	Lapel
Shirt	64.4	84.6	83.2	69.0
Polo	65.3	75.8	75.0	66.7

6.4. Cloth Grasping Experiments

Finally, we evaluate the complete pipeline proposed in this paper in a real robotic experiment in our laboratory, with a WAM arm and a custom-made gripper (see Figure 10).

We use the same experimental setup employed to acquire the dataset for offline evaluation: a table where the garment lays with the robotic arm in one side, and a Kinect camera observing it from a zenithal position (see Figure 1 bottom).

Table 6 shows results on an experiment consisting in randomly laying either a shirt or a polo shirt for a hundred repetitions each, and attempting a grasp on the point suggested by the proposed method. SIFT has been used for the collar detection part, and FINDDD (13 orientations and 43-pixel sided support regions) for the lapel point selection. As shown in the table, on average, a valid lapel point was selected in 80.2% of the cases where the collar was correctly detected (64.9%) when using 128 visual words, and 67.9% when using 1024 visual words (79.1% average correct collar detections). For this experiment we used an extended training set including all relevant images available in the dataset.

As can be seen, using 1024 visual words significantly improves the results of the collar detection step when compared to only 128, but at an slightly higher

Table 7: Overall characterization of grasping failures for the lapel grasping experiments. See text for details.

Error type	Percent.
Mechanical or software error	25.0
Grasp too shallow	19.4
Grasp too deep	11.1
Inverse kinematics error	36.1
Other	8.3

computational cost. This increase in detected collars has an incidence in the performance of the lapel grasping point detector, as it has to deal with harder examples in which valid grasping points must be found.

Furthermore, in some cases where the collar was not considered correctly detected, the lapel classifier still managed to find a valid grasping point if it was present in the considered detection bounding box. For the polo shirt, the lapel classifier was able to find a correct grasping point in 37.1% of the cases where the collar detector failed when using 128 visual words, and 24% when using 1024 visual words.

Some examples of the images and score maps produced in the experiments can be seen in Figure 11. In the failure case shown in the middle-right panel, a wrinkle locally resembles the slope of the shirt lapel and attracts the lapel classifier. Conversely, the "collar probability" of the selected point is low. We have experimented with direct combination of the lapel and collar probabilities, and with forcing the selected point to have a certain minimum collar probability, but none of these two approaches clearly yielded improved results, and a more elaborate methodology will be necessary. In future work we plan to find a way to combine the two scores to avoid this type of errors.

Finally, we also report results on the outcome of the grasp action. A successful grasp was performed 82.1% of the times, and failures were grouped in five categories in Table 7: *mechanical or software error*, for cases where the arm or gripper couldn't execute the instructed action; *grasp too shallow* and *grasp too deep* are cases in which the measured depth was inaccurate and either the gripper didn't reach the garment, or it pushed too hard against the table and failed to close; *inverse kinematics error* corresponds to cases where the detected grasping point falls outside of the working area of the robot; and *other* for miscellaneous errors. Although we have qualitatively assessed the garment hanging step, the methods and technicalities involved are outside the scope of this work, and will be addressed in the future.

7. Conclusions

In this work we have presented a two-stage approach to determine a grasping point in the lapel of a collar of a garment (specifically a shirt or a polo shirt) for hanging purposes. The first stage uses appearance (SIFT) to determine

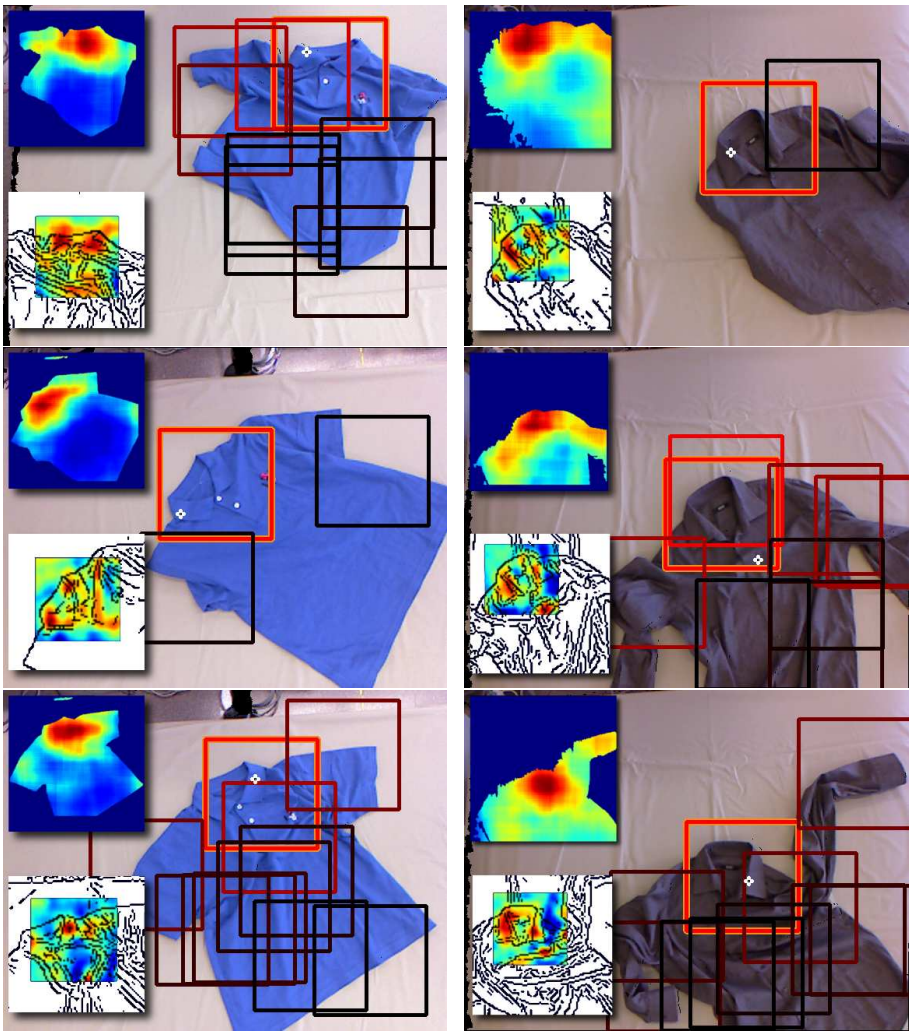


Figure 11: Example results of the proposed perception approach in the experiment of hanging shirts and polos with the robotic arm. Left column corresponds to the blue polo shirt and right column to the shirt. The main figure shows the garment with the detected bounding boxes, organized from red to black with decreasing probability, and a white cross indicating the final selected grasping point. The selected bounding box is highlighted in orange. The two small panels show the **collar** (top) and **lapel** (bottom) probability maps. The figure in the middle-right shows a failure case of the method: the shape of the wrinkle is similar to that of a lapel.

the coarse position of the collar and, in the second stage, depth information (FINDDD) is used to precisely locate a point in the lapel of the collar.

The coarse-to-fine approach shows promising results dealing with the very challenging task of detecting task-oriented grasping points in garments, where the appearance is highly variable, and methods typically resort to long manipulation sequences to guarantee that the garment has been brought to a certain state, such as in [7].

Depth information has been shown to help when dealing with the variable appearance of garments, but state-of-the-art shape descriptors are too computationally expensive to be used in practice in our robotic pipeline. Consequently, we proposed a novel shape descriptor that, taking advantage of the specific properties of our scenario, is near two orders of magnitude faster to compute, while yielding a similar or better performance.

The proposed system has been evaluated both offline with a pre-acquired dataset [36] and online in a robotic garment grasping experiment. Currently the system performs well for mildly deformed garments, with a runtime of only a few seconds. More training data will help improve performance in more difficult scenarios, but generating high quality annotations for this task is very costly, thus it would be desirable to develop a semi-supervised training method.

In future work we would like to explore how to effectively fuse appearance and depth descriptors to optimize the usage of each source of information. We also want to evaluate the proposed approach on other types of clothes and parts, and integrate the method in a planning schema, so that manipulation of the garment can be used to improve robustness. Finally, we would also like to evaluate the proposed FINDDD descriptor in other tasks, such as wrinkle segmentation, or description and recognition of 3D rigid objects.

Acknowledgments

This research is partially funded by the Spanish Ministry of Economy and Competitiveness under project TIN2014-58178-R, by the CSIC project MANIPlus (201350E102), and by the ERA-Net CHISTERA projects ViSen PCIN-2013-047 and I-DRESS PCIN-2015-147. A. Ramisa worked under the JAE-DOC grant from CSIC and FSE. The authors are grateful to the Nvidia donation program for its support with GPU cards.

References

[1] R. B. Rusu, N. Blodow, M. Beetz, Fast Point Feature Histograms (FPFH) for 3D registration, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, 2009, pp. 3212–3217.

[2] M. Bronstein, I. Kokkinos, Scale-invariant heat kernel signatures for non-rigid shape recognition, in: Proc. Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 1704–1711.

[3] F. Tombari, S. Salti, L. D. Stefano, Unique signatures of histograms for local surface description, in: Proc. European Conference on Computer Vision (ECCV), Springer, 2010, pp. 356–369.

[4] A. Ramisa, G. Alenya, F. Moreno-Noguer, C. Torras, Using depth and appearance features for informed robot grasping of highly wrinkled clothes, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 1703–1708.

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

[6] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, P. Abbeel, Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, 2010, pp. 2308–2315.

[7] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, P. Abbeel, Bringing Clothing into Desired Configurations with Limited Perception, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 3893–3900.

[8] S. Miller, M. Fritz, T. Darrell, P. Abbeel, Parametrized Shape Models for Clothing, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 4861–4868.

[9] S. Miller, J. van den Berg, M. Fritz, T. Darrell, K. Goldberg, P. Abbeel, A geometric approach to robotic laundry folding, The International Journal of Robotics Research 31 (2) (2011) 249–267.

[10] S. C. Hidayati, W.-H. Cheng, K.-L. Hua, Clothing genre classification by exploiting the style elements, in: Proc. International Conference on Multimedia (MM), ACM Press, New York, New York, USA, 2012, pp. 1137–1140.

[11] K. Yamazaki, M. Inaba, Clothing Classification Using Image Features Derived from Clothing Fabrics , Wrinkles and Cloth Overlaps, in: Proc. of the International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, Tokyo, Japan, 2013, pp. 2710–2717.

[12] B. Willimon, S. Birchfield, I. Walker, Classification of Clothing using Interactive Perception, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 1862–1868.

[13] B. Willimon, I. Walker, S. Birchfield, Classification of Clothing Using Midlevel Layers, ISRN Robotics 2013 (2013) 1–17.

[14] A. Doumanoglou, A. Kargakos, T.-K. Kim, S. Malassiotis, Autonomous Active Recognition and Unfolding of Clothes Using Random Decision Forests and Probabilistic Planning, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 987–993.

[15] F. Stein, G. Medioni, Structural indexing: efcient 3-D object recognition, IEEE Trans. PAMI 14 (2) (1992) 125–145.

[16] A. E. Johnson, Spin-Images: A Representation for 3-D Surface Matching, Ph.D. thesis, Carnegie Mellon University (1997).

[17] G. Hetzel, B. Leibe, P. Levi, B. Schiele, 3D object recognition from range images using local feature histograms, Proc. Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2001, pp. 394–399.

[18] A. Flint, A. Dick, A. V. D. Hengel, Thrift : Local 3D Structure Recognition, in: Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on, Glenelg, Australia, 2007, pp. 182–188.

[19] G. Flitton, T. P. Breckon, N. Megherbi, A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery, Pattern Recognition 46 (9) (2013) 2420–2436.

[20] G. Flitton, A. Mouton, T. P. Breckon, Object classification in 3D baggage security computed tomography imagery using visual codebooks, Pattern Recognition, Published online, 2015.

[21] B. Steder, R. Rusu, K. Konolige, W. Burgard, Point feature extraction on 3d range scans taking into account object boundaries, in: Proc. International Conference Robotics and Automation (ICRA), IEEE, 2011, pp. 2601–2608.

[22] R. Rusu, Z. Marton, N. Blodow, M. Beetz, Persistent point feature histograms for 3D point clouds, in: Intelligent Autonomous Systems 10: IAS-10, 2008, p. 119.

[23] R. Rusu, Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments, PhD dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October, 2009.

[24] S. Holzer, R. Rusu, M. Dixon, S. Gedikli and N. Navab, Adaptive Neighborhood Selection for Real-Time Surface Normal Estimation from Organized Point Cloud Data Using Integral Images, in: Proc. of the International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, Algarve, Portugal, 2012, pp. 2684–2689.

[25] D. Holz, S. Holzer, R. Rusu and S. Behnke, Real-Time Plane Segmentation using RGB-D Cameras, in: Proc. of the 15th RoboCup International Symposium, Istanbul, Turkey, 2011, pp. 306–317

[26] G. Csurka, C. R. Dance, L. Fan, C. Bray, J. Willamowski, Visual Categorization with Bags of Keypoints, in: ECCV Workshop on Statistical Learning in Computer Vision, 2004, pp. 1–22.

[27] R. G. Cinbis, J. Verbeek, C. Schmid, Image Categorization using Fisher Kernels of Non-iid Image Models, in: Proc. Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2184–2191.

[28] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in large margin classifiers, 1999, pp. 61–74.

[29] H. Lin, C. Lin, R. Weng, A note on Platt's probabilistic outputs for support vector machines, Machine learning 68 (3) (2007) 267–276.

[30] V. Lepetit, P. Fua, Keypoint recognition using randomized trees., IEEE transactions on pattern analysis and machine intelligence (TPAMI) 28 (9) (2006) 1465–79.

[31] G. Yu and J.-M. Morel, ASIFT: An Algorithm for Fully Affine Invariant Comparison, Image Processing On Line, vol. 1, 2011.

[32] M. Villamizar, A. Sanfeliu, F. Moreno-Noguer, Fast Online Learning and Detection of Natural Landmarks for Autonomous Aerial Robots, in: Proc. International Conference on Robotics and Automation (ICRA), IEEE, Hong Kong, China, 2014, pp. 4996–5003.

[33] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[34] A. Klaser, M. Marszalek, C. Schmid, A Spatio-Temporal Descriptor Based on 3D-Gradients, in: Proc. 19th British Machine Vision Conference (BMVC), Vol. 2008, 2008, pp. 275:1–10.

[35] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image Classification with the Fisher Vector: Theory and Practice, International Journal of Computer Vision 105 (3) (2013) 222–245.

[36] A. Ramisa, G. Alenyà, F. Moreno-Noguer, C. Torras, The IRI Clothing Part Dataset (2012).
URL `www.iri.upc.edu/groups/perception/#clothingDataset`

[37] A. Ramisa, G. Alenyà, F. Moreno-Noguer, C. Torras, Learning RGB-D descriptors of garment parts for informed robot grasping, Engineering Applications of Artificial Intelligence 35 (2014) 246–258.

[38] S. Sedai, M. Bennamoun, D. Q. Huynh, Discriminative fusion of shape and appearance features for human pose estimation, Pattern Recognition 46 (12) (2013) 3223–3237.

[39] 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: Proc. International Conference on Robotics and Automation (ICRA), IEEE, Seattle, USA, 2015, pp. 185–192.

[40] C. Zhang, X. Yang, Y. Tian, Histogram of 3D facets: A characteristic descriptor for hand gesture recognition, in: Proc. 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2013, pp. 1–8.