2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Perception for the Manipulation of Socks

Ping Chuan Wang, Stephen Miller, Mario Fritz, Trevor Darrell, Pieter Abbeel

Abstract— **We consider the perceptual challenges inherent in the robotic manipulation of previously unseen socks, with the end goal of manipulation by a household robot for laundry. The task poses challenging problems in modeling the appearance, shape and configuration of these textile items that tend to exhibit high variability in texture, design, and style while being highly articulated objects.**

At the heart of our approach is a holistic model of shape and appearance that facilitates manipulation of those delicate items—starting even from bunched up instances. We describe novel approaches to two key perceptual problems: (i) Inferring the configuration of the sock, and (ii) determining which socks should be paired together.

Robust inference in our model is achieved by strong texture based classifiers that, alone, are powerful enough to solve problems such as inside-out detection. Finally, a reliable prediction of the overall configuration is achieved by combining local cues in a global model that enforces structural consistency.

We perform an extensive evaluation of different feature types and classifiers and show strong performance on each subtask of our approach. Finally, we illustrate our approach with an implementation on the Willow Garage PR2—a general purpose robotic platform.

I. INTRODUCTION

Since Rosie the Robot first debuted on television's "The Jetsons" in 1962, the futuristic image of a personal robot autonomously operating in a human home has captivated the public imagination. Yet, while robots have become an integral part of modern industrial production, their adoption in these less well-defined and less structured environments has been slow. Indeed, the high variability in, for example, household environments, poses a number of challenges to robotic perception and manipulation.

The problem of robotic laundry manipulation exemplifies this difficulty, as the objects with which the robot must interact have a very large number of internal degrees of freedom. This presents a number of unique perceptual challenges. In this work, we examine the perceptual aspects of one particular application: bringing scattered, arbitrarily configured socks into organized pairs.

As many of the difficulties associated with this task are shared with all clothing articles, we believe the strategies developed here will prove useful well beyond their immediate scope.

Socks are extremely irregular, both in shape and appearance. Like all deformable objects, they maintain no rigid structure. Yet, more so than many common articles (such as shirts or pants), they trace no easily-recognizable silhouette, and contour alone offers little guidance. Furthermore, their tubular shape lends itself to highly complex configurations: the sock may be rightside-out, inside-out, or arbitrarily bunched. As it contains no overtly recognizable landmark features—such as buttons or zippers—the perceptual task is quite subtle, residing at the lowest levels of texture.

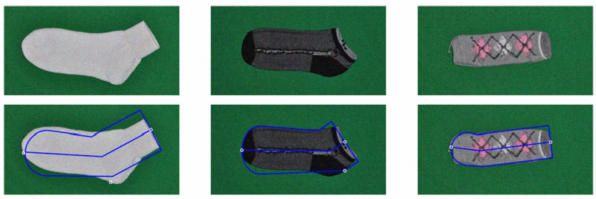
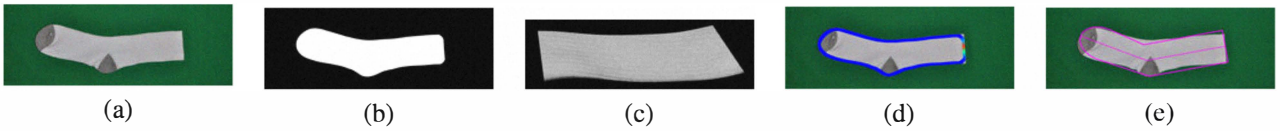


Fig. 1. Given an initial image, we wish to recover the sock configuration.

Our main contributions are as follows:

- *Local texture and shape descriptors for patch recognition:* The core of our approaches hinges on the use of highly discriminative local features for cloth texture. To this end, we examine a variety of texture- and shape-based patch features and choices of kernels. We have found that a combination of Local Binary Pattern (LBP) and shape features, trained with a χ^2 kernel, are well-suited to this task. Our work shows that these cues alone are typically powerful enough to determine whether, for instance, a sock is inside-out.
- *A model-based approach to determine sock configuration:* To reduce noise and enforce structural consistency, we combine the aforementioned descriptors into a global appearance model for socks. This model uses a combination of local texture cues and global contour to infer a basic parse of the sock's configuration, as would be relevant to most robotic manipulation tasks. We use this both as a means of classification and description: classifying whether a sock is flattened, inside-out, or bunched, and determining the location of key features within these configurations.
- *A similarity metric for matching socks:* We developed a similarity score based on a variety of visual cues (texture at different scales, color histogram representations, and size) for pairing socks. Our approach uses this distance metric as input to a matching algorithm to find the set of matches that maximize the sum of the matches' similarity scores. We achieve perfect matching on our database of 100 socks and further show robustness to adding stray socks to the set.
- *Robotic implementation:* To illustrate the effectiveness of our perceptual tools, we implement our approach on the Willow Garage PR2.

Ping Chuan Wang, Stephen Miller, Mario Fritz, Trevor Darrell, and Pieter Abbeel are with the the University of California at Berkeley. E-mail: w50922@berkeley.edu, {sdmiller,mfritz,trevor,pabbeel}@eecs.berkeley.edu.



(a) (b) (c) (d) (e)

Fig. 2. (a) We begin with an image of a sock in one of 8 poses. (b) The background is subtracted to obtain a mask of the sock. (c) A local texture classifier is used to determine landmark patches. (d) Visualization of the response of landmark filters on a sock. (e) These local features are then combined in a global shape model, which determines the configuration of the sock.

II. RELATED WORK

We present the related work along three axes: robotic manipulation of cloth, grasping, and visual perception of textures and material.

A. Robotic manipulation of cloth

The current state of the art in robotic cloth manipulation is still far removed from having a generic robot perform tasks such as laundry. In the context of robotic laundry, past research has mostly considered shape cues for perception, enabling tasks such folding polygonal shapes from a shape library [1], spreading out a clothing article and classifying its category [2], [3], [4], [5], detecting and removing wrinkles [6], [7], and folding previously unseen towels starting from arbitrary configurations [8]. While shape cues have the benefit of being robust across appearances and are a natural choice in the aforementioned tasks, shape cues only provide very limited information for the purpose of arranging socks. In this paper we do not restrict ourselves to shape cues, but also study, and in fact get most leverage from, cues based on texture and color.

Socks require rather different manipulations than the grasp and tugging strategies exhibited in prior cloth manipulation work. Besides the mere scale and tubular structure of socks, their structure necessitates particularly complex motions, such as flipping and bunching. While the emphasis of this work is on perception, we also integrated our perception algorithms onto a general purpose robot. This work is the first to perform such manipulation primitives with a general purpose robot.

B. Grasping

Our aim is to provide a basic parse of the sock, such that deft manipulations may be performed. The key to many of these manipulations is an accurate initial grasp. Yet while traditional grasp planning is done by reasoning about 3-D configurations of gripper and object, these are often hard to obtain for real-world objects—in particular for the thin layered structure of socks. More recent approaches have aimed at relaxing this often difficult to meet assumption by inferring grasp points for unseen objects from local statistics [9], [10] or parallel structures [11]. However, these works aim only to obtain an arbitrary grasp of the object, mostly for picking it up. Likewise, we wish to enable fine manipulations such as flipping or pairing, which are informed by the topology of the sock rather than broad spatial reasoning.

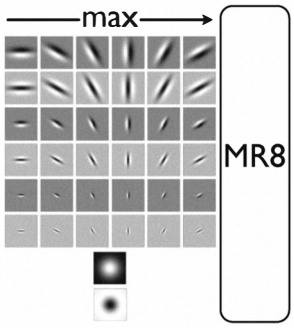


Fig. 3. The MR8 filter bank consists of 6 gaussian derivative and 2 blob filters. A maximum operations is performed over different orientation variants in order to achieve robustness with respect to rotations.

C. Visual Perception of Texture and Materials

Recognition of textures and materials have received wide attention in computer vision. State-of-the-art methods as evaluated on the Curet or KTH-TIPS database [12] include filter- and texton-based techniques like MR8 features [13] or Local Binary Patterns (LBP) [14] as well as MRF-based methods [15]. More recent approaches have further included shape-based techniques like edge maps and curvature [16].

Here we apply these descriptors to robotic manipulation tasks. In particular, we show how to leverage micro-texture in order to classify between different sides of fabric and adapt robotic grasp strategies depending on material properties. We additionally incorporate our descriptors into a broader global cloth shape model, which builds upon [17]. This earlier work is based on shape alone, and does not allow for more complex articulations including flipped and spread-out configurations.

III. METHODS

The following are the key components in our approach:

- Extraction of appearance features;
- Learning a patch classifier;
- Specification of a global model for reconstructing sock configuration;
- A strategy for matching alike socks based on the aforementioned descriptors.

A. Appearance Features

In order to understand the configuration of a sock, our approach relies on pinpointing the location of key landmarks. To identify these landmarks, we study canonical texture cues in a classification framework.

Two categories of features are of interest to us: those based on texture, and those based on local shape.

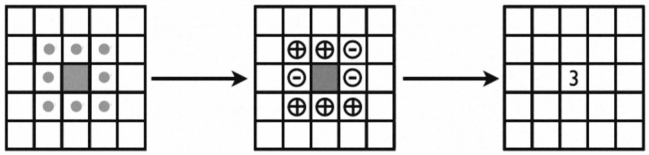


Fig. 4. LBP features are computed by taking a neighborhood (gray circles) around each pixel (gray square) and computing the difference of each neighborhood pixels to the center pixel. The sign of this difference determines a binary value for each neighborhood pixel. The concatenated sequence of binary values is mapped to a unique pattern id.

(a) (b) (c)

Fig. 5. We extract the local shape by using a HOG respresentation. (a) An image of a sock. The blue region indicates a segment along the contour. (b) A mask of a segment along the contour is computed, and rotated such that it is upright. (c) For each segment mask, gradient information is binned into local 9 dimensional orientation histograms. Each bin is visualized with small line segments that have a length according to the accumulated gradient strength.

1) Texture: For texture representation we consider two of the most popular texture descriptors: the MR8 filter bank [13] and Local Binary Patterns (LBP) [14]. Figure 3 and Figure 4 illustrate the basic concepts of those approaches.

MR8 features are computed by convolving the images with a set of filters. For each pixel the maximum is computed over groups of rotated versions of these filters in order to obtain a rotation invariant representation. This results in an 8-dimensional feature space which is vector quantized by k-means. The final descriptor is a histogram that counts matches to the individual codebook vectors of the quantized space.

LBP features are computed by considering the difference in grayscale value between each pixel and its 8 neighbors. Looking at the sign of these differences, we obtain a binary vector. As there is only a limited set of distinct binary patterns, each binary pattern is mapped to a unique pattern id. The final feature is a histogram over these pattern ids. For more details on both methods we refer to the given references.

Both features are extracted at a fine scale as we intend to capture the micro-texture of the fabric, which seems most appropriate for our task.

2) Shape: Because some of the landmarks of a sock, such as the toe and the heel, are additionally related to the local shape, we augment our texture cues to include shape features. The local shape along the contour of a sock can be captured by a Histogram of Oriented Gradients (HOG) [18] computed on the mask of a local contour region split into 2 cells. Figure 5 illustrates the basic concept of this approach.

B. Classifier Training

The solution that we propose for the classification task follows the standard literature on texture classification (e.g., [19], [12]) and trains a discriminative SVM classifier on those texture features. Given a labelled training set with K classes

$$\{[f_1^p, 1] \mid p = 1, \ldots, N_1\} \cup \cdots \cup \{[f_K^p, K] \mid p = 1, \ldots, N_K\} \tag{1}$$

with N_i instances of label i, we seek to train a classifier that correctly predicts the associated class label:

$$f_1^p \rightarrow 1 \quad ; \quad \ldots \quad ; f_K^p \rightarrow K \tag{2}$$

Here f_i^p is the feature vector extracted from a single patch p whose correct label is i. The standard SVM approach [20] learns a classifier by finding appropriate weight vectors w_i that attempt to ensure that for all p we have $w_i^\top f_i^p > w_{j \neq i}^\top f_i^p$. Note that while the classifier is linear in w_i the features are a nonlinear function of the image pixels. The weight vector w_i is determined by solving the following (L2-loss) optimization problem:

$$\begin{aligned} \min_{w,\xi \geq 0} \quad & \sum_{i=0}^{K} \frac{1}{2}\|w_i\|_2^2 + C_i\|\xi_i\|_2^2 \\ \text{s.t.} \quad & \forall p, i, j \neq i : w_i^\top f_i^p \geq w_j^\top f_i^p + 1 - \xi_i^p \end{aligned}$$

In addition, we train the SVM classifier with four different types of kernels: linear, degree-3 polynomial, radial basis function (RBF), and χ^2. Following ([21], [22]), we define the χ^2 kernel as: $\chi^2(x, y) = \exp(-\gamma * \sum_i (x_i - y_i)^2/(x_i + y_i))$, and define the other kernels according to their standard definition. The applications of the χ^2 kernel to texture classification are explored in [23].

C. Global Model

To infer global structure, we incorporate each local feature into a holistic parametrized shape model. In [17] we described the foundation of a parametrized global model using a contour-based approach. Here we extend this method to include local feature cues.

To determine the configuration of an article of clothing, we first define a parametrized model associated with that article. We then frame the task of determining global structure as a numerical optimization problem, whose objective function captures the "goodness of fit" of the model, and whose constraints reflect our *a priori* knowledge of the article's possible variations. This entails three main components: a framework for defining models, a means of determining model fit, and an efficient method for determining good parameters in this setting.

1) Model Definition: A model is, essentially, a parametrized representation of a given shape. As such, it requires a few key components:

- A *contour generator* $M_{CG} : \{P \in \mathbb{R}^p\} \rightarrow \{C \in \mathbb{R}^{2 \times c}\}$ which takes a set of scalar parameters P as input, and returns the contour which would be observed were the model physically present with these parameters.
- A set of *feature detectors* $M_{FD} : \{D_i : \{P \in \mathbb{R}^p, I \in \mathbb{R}^{n \times m \times 3}\} \rightarrow \{r_i \in \mathbb{R}\}\}^K$.

Each detector D_i takes the parameters and image as input, and outputs the response of the patch to label i at its current predicted location.

- A *legal input set* $M_{\mathcal{L}} \subseteq \mathbb{R}^p$ which defines a set of parameters over which M is said to be in a legal configuration.

2) Model Cost: Our model cost, which is a function of the parameters P, the observed image I and the observed contour C, is given by a weighted sum of shape- and appearance-related penalties:

$$\mathcal{C}(P, I, C) = \beta(\mathcal{C}_S(P, C)) + (1 - \beta)(\mathcal{C}_A(P, I))$$

The shape cost is given by

$$\mathcal{C}_S(P, C) = (\alpha)\bar{d}(M_{CG}(P) \to C) + (1-\alpha)\bar{d}(C \to M_{CG}(P)).$$

with $\bar{d}(A \to B)$ the average nearest-neighbor distance from contour A to contour B.

The appearance cost is given by

$$\mathcal{C}_A(P, I) = \begin{bmatrix} \alpha_1 & \dots & \alpha_K \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_K \end{bmatrix}.$$

Here r_i is the response[1] of feature detector i at the predicted feature location, and the weights dictate the importance given to various landmarks. For instance, in determining the configuration of a sock, we may have local detectors for ankles, toes, and generic patches. These are given respective weights of 0.5, 0.4, and 0.1: indicating that it is most important that the model correctly predict the seam, and comparitively unimportant to predict generic patches. In the limit where β approaches 1, this cost function is identical to that defined in [17].

3) Parameter Fitting: To ensure that the optimization begins in a reasonable initial position, we use Principle Component Analysis to infer the approximate translation, rotation, and scale of the observed contour. The details of this procedure may be found in [17].

We then locally optimize the parameters of our model via a numerical coordinate descent algorithm. To guide the optimization, this is often done in multiple phases, which allow varying degrees of freedom. As we do so, we consider only legal configurations, ensuring that the model will not converge on an inconsistent state.

D. Matching

When considering texture thus far, we limited ourselves to micro-texture, which attempts to capture the textile structure of the socks without regard to the broad design of the sock itself. For matching, however, the particular design is exactly what is being matched: therefore we construct our feature vector with a combination of micro texture, macro texture, width, height and color features.

As LBP will show to be the better performer in our pure texture-based experiments, we base our micro texture feature on it, and subsequently compute the macro texture feature by down-scaling the image by a factor of two before extracting the LBP features. Color features are obtained by computing a hue histogram in HSV space with 19 bins, and we have an additional "non-color" bin that collects all pixels with low value or saturation.

We investigate the use of these cues both individually as well as in combination by simply concatenating them together into a feature vector of increased dimensionality. We also investigated learning the weights for cue combination from data in an optimization framework—but the naive strategy of concatenation turns out to be sufficient for our task at hand.

For the actual matching we look at a greedy as well as an optimization-based scheme. For the greedy matching we simply score all possible pairs (s_i, s_j) by our feature distance function $d_{\chi^2}(s_i, s_j) = \sum_i (x_i - y_i)^2/(x_i + y_i)$ and accept successively the best ranked pair. After accepting a pair, we remove the involved socks from further consideration. In the case of stacked features, the distance function is simply the sum of the individual feature distances.

More interestingly, we also propose a optimization scheme that seeks to minimize a global matching score across all pairs. This can be seen as finding a permutation $\mathcal{P}$ such that:

$$\min_{\mathcal{P}} \sum_i d_{\chi^2}(s_i, s_{\mathcal{P}(i)}) \tag{3}$$

The problem of finding such a permutation is known as the minimum cost perfect matching problem. Efficient algorithms exist to find the exact solution: we used the algorithm proposed in [24].

To handle stray socks we start with the lowest scoring (best) pairs and work our way up until the cost exceeds the maximum cost in which the algorithm considers indicative of a proper match. In case of an odd number of socks in the set, we introduce a "fake sock" which has equal similarity with all socks. The true sock matched to the fake sock is considered a stray sock.

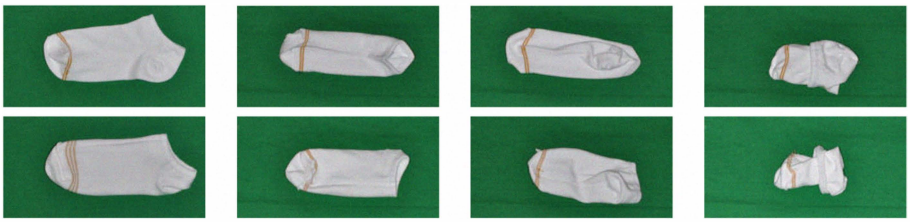
IV. EXPERIMENTS

1) Dataset: We test our approach on 800 images, corresponding to 100 socks laid in 8 canonical configurations, as detailed in Figure 6. The images were taken on a 12.3-megapixel Nikon D90 camera with a 35mm Nikon DX lens, using an external Sigma Macro ring flash. The photos were taken from a birds-eye perspective against a green tablecloth, allowing us to locate the sock contour via simple color segmentation.[2]

Each image was then labelled in two ways. To train the proper feature classifiers, the sock image was hand-segmented by microtexture class: opening, heel, toe, or other for non-bunched models, and inside or outside for bunched

[1]The response is given by $w^\top f$ in the appropriate kernel, as discussed in Sec. III-B

[2]As the manipulation context we are considering will be that of a sock on the floor or other fixed background, we do not consider segmentation to be a particularly relevant problem. However, background subtraction and color segmentation are a well-studied problem in computer vision, and we invite interested readers to look further (see e.g. [25]).



(a) (b) (c) (d)

Fig. 6. There are 4 canonical modes the sock may be in: (a) Sideways, (b) Heel Up, (c) Heel Down, or (d) Bunched. Additionally, the toe may be rightside-out (top) or inside-out (bottom) for each configuration. In conjunction with 2D rotation and reflection, this suffices to capture all reasonable sock configurations.

models. Additionally, to provide a ground truth comparison for all configuration estimates, the precise locations of all landmark points was hand-annotated.

We also supplement this dataset with a second one, which shows all pairs of sock correctly flipped in a non-bunched configuration in order to evaluate our sock matching algorithm.

A. Inside-Out Classification

We first investigate purely texture-based classification of inside-out vs rightside-out socks.

First we compare the 2 popular texture descriptors—LBP and MR8 [3]—and a range of different kernel choices for the SVM classifier: linear, polynomial (degree 3), RBF, and χ^2. [4]

We compute the accuracy over 5 random splits of the dataset into equal-sized training and test sets. The SVM parameters are determined by cross-validation on the training set. Table I presents mean accuracy and standard deviation of the different splits.

LBP in combination with the χ^2 kernel shows the best performance at $96.6 \pm 1.82\%$, while the best MR8 result lags behind at $87.8 \pm 2.77\%$. [5]

With this choice of texture descriptor and kernel, we investigate how important resolution of the sock images is. Figure 7 shows the result of the classifier training and testing on images downscaled to various resolutions. The graph shows a very graceful degradation in the performance of the texture descriptor as we downscale the image resolution. Furthermore, the performance of the texture descriptor does not benefit from increased resolution after 4-megapixels, so further increase in the resolution of the camera is unlikely to increase the performance of our results.

B. Recognizing Sock Configuration

Our primary goal is to provide the perception necessary to allow a robot to manipulate socks. To that end, we desire to perceive enough about the structure of the sock to perform such motions as:

- Flipping an inside-out sock;
- Pairing socks at the ankle;
- Bringing a bunched sock into a planar configuration.

These manipulations require a general understanding of the configuration of a sock, including for instance its orientation, the locations of the sock opening, and whether or not it is inside-out.

In the sections that follow, we demonstrate two approaches to gain such an understanding: 1) Using only local features, we attempt to recover the sock's configuration 2) We integrate these local features into a global model which considers both appearance and contour information.

We evaluate the accuracy of these approaches via two error metrics, giving a qualitative and quantitative measure of accuracy per landmark prediction. The qualitative measure is computed by comparing the predicted location to the microtexture-labelled image. If the 10 pixel (roughly 1 mm) neighborhood surrounding the prediction contains the proper label, it is deemed a success. The quantitative measure is the distance from the predicted location to the precise, hand-annotated landmark.

Feature	Linear Kernel	Poly Kernel	RBF Kernel	χ^2 Kernel
MR8	$85.2 \pm 2.28\%$	$85.4 \pm 1.95\%$	$86.6 \pm 3.13\%$	$87.8 \pm 2.77\%$
LBP	$93.8 \pm 3.35\%$	$94.2 \pm 3.63\%$	$95.8 \pm 2.39\%$	$\mathbf{96.6 \pm 1.82\%}$

TABLE I

A COMPARISON OF THE PERFORMANCE OF MR8 AND LBP DESCRIPTORS IN COMBINATION WITH VARIOUS KERNELS.

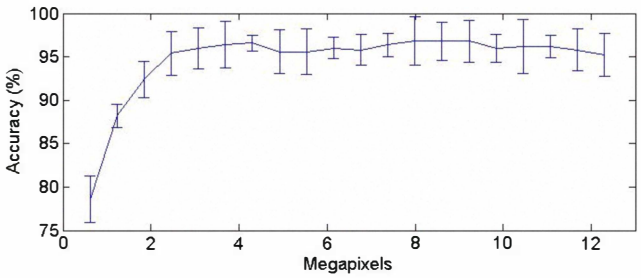


Fig. 7. Accuracy of using LBP in combination with χ^2 kernel for the inside-out vs. rightside-out classification on images from the dataset downscaled to various resolutions.

[3] We trained the MR8 descriptor with a texton dictionary consisting of 256 cluster centers.

[4] All classifiers were trained using the LIBSVM software package [26], modified to include the χ^2 kernel.

[5] In this work, patch rotation is normalized with respect to the contour it borders. To ensure a fair comparison between descriptors, we additionally tested modified versions of each descriptor: a rotationally variant MR8, and invariant LBP as discussed in [27]. In the end, the rotationally variant LBP with the χ^2 kernel still shows the best performance.

		Appearance		Contour		Model Known		Model Unknown	
Configuration	Landmark	Qual (%)	Quant (cm)	Qual (%)	Quant (cm)	Qual (%)	Quant (cm)	Qual (%)	Quant (cm)
Side	Opening	$98.8 \pm 0.4\%$	1.04 ± 0.67	$92.0 \pm 1.41\%$	1.65 ± 5.43	$98.0 \pm 0.63\%$	0.68 ± 0.64	$96.0 \pm 1.9\%$	0.85 ± 1.31
	Heel	$85.8 \pm 1.72\%$	2.26 ± 2.25	$83.6 \pm 1.62\%$	1.63 ± 2.61	$85.8 \pm 2.64\%$	1.98 ± 2.30	—	—
	Toe	$93.2 \pm 3.54\%$	1.67 ± 1.76	$95.0 \pm 1.41\%$	1.65 ± 5.58	$100.0 \pm 0.0\%$	0.91 ± 0.61	$99.6 \pm 0.5\%$	1.08 ± 1.27
Heel Up/Down	Opening	$91.5 \pm 1.61\%$	1.87 ± 4.17	$71.1 \pm 1.43\%$	6.28 ± 10.9	$94.7 \pm 1.03\%$	1.40 ± 4.61	$93.8 \pm 0.5\%$	1.36 ± 3.42
	Toe	$92.8 \pm 1.17\%$	2.21 ± 4.12	$72.7 \pm 1.57\%$	6.49 ± 10.9	$97.0 \pm 0.54\%$	1.91 ± 4.58	$96.1 \pm 0.4\%$	1.67 ± 3.42
Bunched	Opening	$44.6 \pm 2.06\%$	2.25 ± 1.69	$38.0 \pm 3.03\%$	2.84 ± 1.95	$73.0 \pm 4.47\%$	1.46 ± 1.60	—	—

TABLE II

THE PERFORMANCE OF THE APPEARANCE, CONTOUR ONLY, AND GLOBAL MODELS FOR EACH CONFIGURATION AND LANDMARK.

Detection via Appearance Features

We first attempt to determine the configuration using local feature detectors. In this case, it is assumed that the general configuration—sideways, flattened, or bunched—is given. For each of these configurations, we train a set of detectors to locate particular sock regions.

- Side View: For the side-view configuration, we trained classifiers for four texture categories: the opening, the toe, the heel, and generic patches. As each of these features lie on the contour, this model does not consider the response of interior patches.
- Heel Up/Down View: When the heel is entirely vertical, we make no attempt at finding it. Rather, we simply observe that the heel is not in a relevant location for grasping, and search only for opening, toe, and generic patches. This also does not consider the response of interior patches.
- Bunched: For the bunched configuration, we trained classifiers for the opening, and generic patches. This model only considers the response of the interior patches, as the opening does not lie on the contour.

Each landmark point is then determined to be the center of the maximally-responding patch for its corresponding detector. Table II details the results of this approach.

Detection via a Global Model

We then integrated the above feature detectors into a global model, as detailed in III-C. To perform a global classification, three separate models were considered. These models are shown in Figure 8. Appearance scores were computed in the following way:

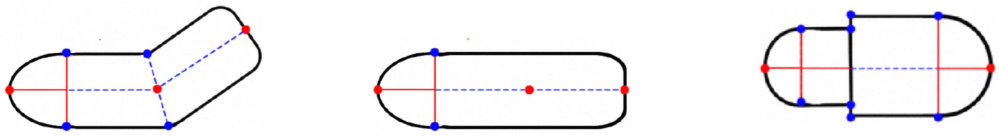
- The Side-View Model computes the appearance cost using Opening, Heel, Toe, and Generic responses, with respective weights of 0.4, 0.35, 0.2, and 0.05. The location of the first three landmarks can be inferred from the skeletal structure of the model. The Generic responses, in all models, are a weighted sum of the response of the five remaining polygon points. This model was run with 4 separate initializations, corresponding to every combination of heel and toe directions.
- The Heel-Up/Down Model computes the appearance cost using Opening, Toe, and Generic responses, with respective weights of 0.5,0.4, and 0.1. The remaining polygon points, as well as the top- and bottom- center of the sock, are used to compute the Generic score. This model was run with 2 separate initializations: one in which the toe was left of the opening, and its mirror.
- The Bunched Model computes the appearance cost using local inside and outside patch responses. The appearance response is then computed as the sum of the average inside-out response on one side of the predicted opening and average rightside-out response on the other. To keep the score continuous despite the discrete step size between patch locations, a low-weighted term is added to this which penalizes the distance from the opening to those patches whose responses do not fit our hypothesis. This model was run with 2 separate initializations; one in which the inside-out half was presumed to be on the left, and the other on the right.

For computational efficiency, patch responses are not recomputed precisely at each pixel. Rather, a discrete set of patch responses are precomputed, and the response at a given point is given by bilinear interpolation between the responses of its neighboring patches.

The results of this approach are tabulated in Table II. We consider three cases. As a point of comparison, we first consider the case where β is set to 1—analogous to the pure shape-based approach of [17]. The latter two follow the global model approach outlines above: in the former, the correct model class is known a priori; in the latter, it is not.[6]

As can be seen, the global model improves significantly on the baseline approach in most areas. While the texture and curvature features of the Side View landmarks are fairly telling, their precise location is rendered ambiguous by texture alone. Thus while the global model does little to improve the qualitative accuracy of our predictions, it yields far more precise landmarks. In dealing with the fairly homogenous textures of the Heel Up/Down View, the contour fit proved crucial to gaining high qualitative results. The Bunched configuration, which could not be dealt with by looking for a single local feature, was handled with reasonable accuracy by our Model. While the inherent discrete nature of the approach made it unlikely that the seam would fall precisely in the slim labeled area (yielding lower qualitative results), it remained

[6]In the Model Known case, only the desired feature set (Side View, Heel Up/Down, Bunched) is given: orientation and parity are always unknown. In the Model Unknown case, we consider both Side View and Heel Up/Down models simultaneously on all non-bunched configurations, and choose the maximally scoring configuration between the two. As the recognition task and required manipulations are fairly distinct for Bunched and Non-Bunched cases, we do not include Bunched Models in the latter evaluation.



(a) (b) (c)

Fig. 8. (a) The Side View Model is parametrized by the toe radius, the sock width, the toe, the opening, and a joint. Its only constraint is that the heel be convex. (b) The Heel Up/Down Model is parametrized by the toe radius, the sock width, the toe, the opening, and a joint (c) The Bunched Model is parametrized by two toe radii, two sock-half widths, the distance of the opening, and both end points

Configuration	Classification Accuracy
Bunched	$81.0 \pm 3.81\%$
Non-Bunched	$92.8 \pm 1.37\%$

TABLE III
BUNCHED CLASSIFICATION

more or less as precise as all other configurations. Example successes and failures are shown in Figure 9.

Finally, we consider the problem of distinguishing between bunched and non-bunched socks. To do this, we use an approach identical to Section IV-A to train a bunched-vs-non-bunched classifier, using LBP and Shape features and the χ^2 kernel. The classification results are given in Table III.

C. Sock Pairing

Table IV shows the evaluation of our pairing algorithm. Both the greedy and optimized matching strategies are performed on different feature types. We vary the number of socks in the set and present the mean accuracy and standard deviation over 5 random splits of the full dataset.

With a few exceptions, the optimization approach outperforms the greedy strategy and also shows lower variance across the different runs. The MicroLBP feature is the best performing single cue and its performance only drops by 4% when scaling from 10 to 100 socks. Combining all cues in single feature vector yielded perfect matching for all investigated sets of socks for the greedy as well as the optimization method. To further challenge our greedy matching algorithm we also investigate sets to which we added single stray socks. Figure 10 shows a precision recall evaluation of the combined cues approach where we successively detected pairs of socks according to the matching score while we record recall and precision with respect to the correct pairs:

$$\text{precision} = \text{tp}/(\text{tp} + \text{fp}) = \frac{\#\text{correctly matched}}{\#\text{predicted matches}} \quad (4)$$

$$\text{recall} = \text{tp}/(\text{tp} + \text{fn}) = \frac{\#\text{correctly matched}}{\#\text{pairs in database}} \quad (5)$$

While our method performs without errors for the case of no stray socks, we can observe a very graceful degradation up to the case where 50 stray socks are mixed with 50 pairs—for which we still achieve 96.1% recall at a precision of 98.0%.

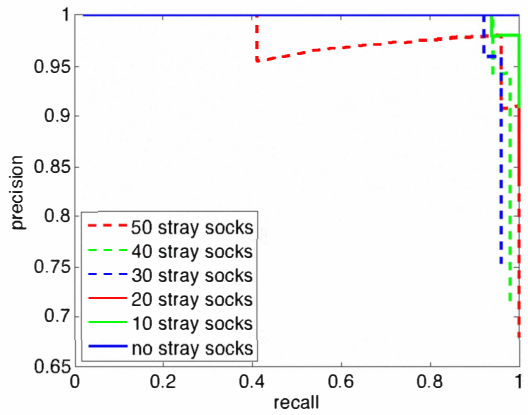


Fig. 10. We evaluate our matching performance on a set of 50 pairs of socks. The precision-recall plot shows how precision degrades from the most confident pair to the least confident one as we successively add up to 50 stray socks.

D. Robotic Implementation

To illustrate the power of these perceptual tools, we implemented it on the Willow Garage PR2 robot. Videos of the functioning system are available at:

`http://rll.berkeley.edu/2011_IROS_socks`

V. CONCLUSIONS

We considered the problem of equipping a robot with the perceptual tools for reliable sock manipulation. We framed this as a model-based optimization problem, whose primary components are local texture descriptors and a contour-matching strategy. We examined in detail the strength of individual textural features, and augmented them with local shape cues. These tools enabled us to reliably recover the configuration of potentially bunched socks. Additionally, we proposed a feature-matching algorithm for sock pairing.

REFERENCES

[1] N. Fahantidis, K. Paraschidis, V. Petridis, Z. Doulgeri, L. Petrou, and G. Hasapis, "Robot handling of flat textile materials," *Robotics & Automation Magazine, IEEE*, vol. 4, no. 1, pp. 34–41, Mar 1997.

[2] F. Osawa, H. Seki, and Y. Kamiya, "Unfolding of massive laundry and classification types by dual manipulator," *JACIII*, vol. 11, no. 5, pp. 457–463, 2007.

[3] K. Hamajima and M. Kakikura, "Planning strategy for task of unfolding clothes," in *Proc. ICRA*, vol. 32, 2000, pp. 145–152.

[4] Y. Kita, F. Saito, and N. Kita, "A deformable model driven visual method for handling clothes," in *Proc. ICRA*, 2004.

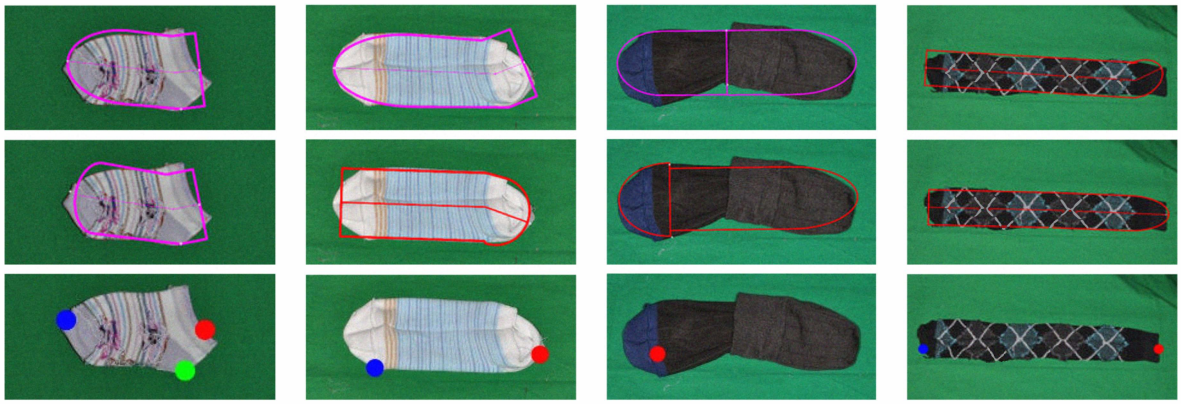


Fig. 9. Example successes and failures of the global (top), contour (middle), and appearance (bottom) models. For the appearance models, the colored circles represented the predicted landmark locations: red indicates the opening location, blue indicates the toe location, and green indicates the heel location.

Feature	n=10	n=20	n=30	n=40	n=50	n=60	n=70	n=80	n=90	n=100
MicroLBPGreedy	96.0 ± 8.9%	96.0 ± 5.5%	94.4 ± 5.9%	96.0 ± 2.2%	95.2 ± 6.6%	93.0 ± 2.1%	94.6 ± 3.9%	93.2 ± 1.6%	91.8 ± 1.8%	92.0%
MicroLBPOpt	98.0 ± 4.5%	98.0 ± 4.5%	98.0 ± 3.0%	97.5 ± 3.5%	96.0 ± 2.8%	95.3 ± 2.5%	94.0 ± 1.2%	95.2 ± 1.0%	94.4 ± 0.0%	95.0%
MacroLBPGreedy	96.0 ± 8.9%	94.0 ± 8.9%	93.0 ± 4.9%	93.0 ± 5.7%	92.0 ± 8.6%	89.6 ± 4.7%	90.6 ± 3.9%	89.2 ± 2.2%	87.0 ± 2.8%	87.0%
MacroLBPOpt	96.0 ± 8.9%	98.0 ± 4.5%	93.3 ± 4.7%	94.0 ± 4.2%	92.8 ± 7.7%	92.7 ± 4.2%	92.6 ± 3.7%	91.5 ± 2.4%	89.8 ± 2.7%	89.0%
SizeGreedy	56.0 ± 18.2%	48.0 ± 9.1%	29.6 ± 6.5%	30.8 ± 4.6%	27.6 ± 3.8%	19.6 ± 3.8%	16.8 ± 3.4%	17.8 ± 2.4%	13.6 ± 1.5%	13.0%
SizeOpt	46.0 ± 11.4%	52.0 ± 6.9%	33.3 ± 7.0%	39.0 ± 7.4%	24.2 ± 3.3%	23.2 ± 6.5%	20.7 ± 5.6%	18.8 ± 5.1%	14.7 ± 1.4%	14.5%
ColorGreedy	92.0 ± 11.0%	96.0 ± 5.5%	98.6 ± 3.1%	94.8 ± 7.9%	93.6 ± 3.3%	94.4 ± 1.9%	90.4 ± 1.5%	92.0 ± 2.5%	91.2 ± 1.3%	91.0%
ColorOpt	96.0 ± 8.9%	98.0 ± 4.5%	98.7 ± 3.0%	94.8 ± 7.8%	88.6 ± 2.4%	91.3 ± 3.8%	89.3 ± 2.3%	91.0 ± 1.3%	89.0 ± 1.8%	89.0%
AllGreedy	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0%
AllOpt	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0%

TABLE IV

ACCURACY OF SOCK PAIRING ALGORITHM FOR DIFFERENT FEATURE TYPES AND SOCK SET SIZES AS WELL AS THE GREEDY VS. THE GLOBAL OPTIMAL MATCHING STRATEGY. WE ACHIEVE PERFECT PAIRING FOR THE WHOLE DATASET BY COMBINING ALL THE CUES.

[5] M. Cusumano-Towner, A. Singh, S. Miller, J. F. OBrien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *ICRA*, Berkeley, California, 2011.

[6] K. Yamakazi and M. Inaba, "A cloth detection method based on image wrinkle feature for daily assistive robots," in *IAPR Conf. on Machine Vision Applications*, 2009, pp. 366–369.

[7] H. Kobori, Y. Kakiuchi, K. Okada, and M. Inaba, "Recognition and motion primitives for autonomous clothes unfolding of humanoid robot," in *Proc. IAS*, 2010.

[8] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2010.

[9] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. J. Rob. Res.*, vol. 27, no. 2, pp. 157–173, 2008.

[10] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robot. Auton. Syst.*, vol. 58, no. 4, pp. 362–377, 2010.

[11] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger, "A strategy for grasping unknown objects based on co-planarity and colour information," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 551–565, 2010.

[12] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the significance of real-world conditions for material classification," in *ECCV*, Prague, Czech Republic, 2004.

[13] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Comput. Vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[14] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51 – 59, 1996.

[15] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?" in *CVPR*, vol. 2, June 2003, pp. 691–698.

[16] C. Liu, L. Sharan, E. Adelson, and R. Rosenholtz, "Exploring features in a bayesian framework for material recognition," 2010.

[17] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Parametrized shape models for clothing," in *ICRA*, Berkeley, California, 2011.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[19] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," in *ICCV*, 2005.

[20] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1996.

[21] O. Chapelle, P. Haffner, and V. Vapnik, "Svms for histogram-based image classification," 1999.

[22] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, "Spectral partitioning with indefinite kernels using the Nyström extension," *Computer VisionECCV 2002*, pp. 51–57, 2002.

[23] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," 2005.

[24] Z. Galil, "Efficient algorithms for finding maximum matching in graphs," *ACM Computing Surveys (CSUR)*, vol. 18, no. 1, pp. 23–38, 1986.

[25] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice-Hall, 2003.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[27] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," *Computer Vision-ECCV 2000*, pp. 404–420, 2000.