

## Research Article

# Classification of Clothing Using Midlevel Layers

Bryan Willimon, Ian Walker, and Stan Birchfield

*Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA*

Correspondence should be addressed to Bryan Willimon; rwillim@clemson.edu

Received 31 December 2012; Accepted 26 January 2013

Academic Editors: K. K. Ahn, L. Asplund, and R. Safaric

Copyright © 2013 Bryan Willimon et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a multilayer approach to classify articles of clothing within a pile of laundry. The classification features are composed of color, texture, shape, and edge information from 2D and 3D data within a local and global perspective. The contribution of this paper is a novel approach of classification termed L-M-H, more specifically LC-S-H for clothing classification. The multilayer approach compartmentalizes the problem into a high (H) layer, multiple midlevel (characteristics (C), selection masks (S)) layers, and a low (L) layer. This approach produces “local” solutions to solve the global classification problem. Experiments demonstrate the ability of the system to efficiently classify each article of clothing into one of seven categories (pants, shorts, shirts, socks, dresses, cloths, or jackets). The results presented in this paper show that, on average, the classification rates improve by +27.47% for three categories (Willimon et al., 2011), +17.90% for four categories, and +10.35% for seven categories over the baseline system, using SVMs (Chang and Lin, 2001).

## 1. Introduction

Sorting laundry is a common routine that involves classifying and labeling each piece of clothing. This particular task is not close to becoming an automated procedure. The laundry process consists of several steps: handling, washing, drying, separating/isolating, classifying, unfolding/flattening, folding, and putting it away into a predetermined drawer or storage unit. Figure 1 gives a high-level flow chart of these various steps. In the past, several bodies of work have attempted at solving the tasks of handling [1–8], separating/isolating [8–12], classifying [6, 9, 11–15], unfolding/flattening [14, 16], and folding [17] clothes. Figure 1 gives a flow chart of the various areas.

A robotic classification system is designed to accurately sort a pile of clothes in predefined categories, before and after the washing/drying process. Laundry is normally sorted by individual, then by category. Our procedure allows for clothing to be classified/sorted by category, age, gender, color (i.e., whites, colors, darks), or season of use. The problem that we address in this paper is grouping isolated articles of clothing into a specified category (e.g., shirts, pants, shorts, cloths, socks, dresses, jackets) using midlevel layers (i.e., physical characteristics and selection masks).

Previous work on classifying clothing [6, 11–14] used dual manipulators to freely hang each article of clothing at two grasp points to reveal the overall shape of each article of clothing. In contrast to previous work, our approach is tested on an unorganized, unflattened article for the purpose of classifying each individual piece of clothing. The database of clothing, that we use in this paper, consists of individual articles placed on a table by a single robotic manipulator.

Our approach uses a set of characteristics, a local/global histogram for each article, and low-level image/point cloud calculations in order to accomplish this task. The proposed method can be seen as a particular application of the paradigm of interactive perception, also known as manipulation-guided sensing, in which the manipulation is used to guide the sensing in order to gather information not obtainable through passive sensing alone [18–22]. The articles of clothing that we use in this paper have been manipulated in a pre-defined set of movements/operations in order to gather more information for our approach. Each article of clothing is initially placed on a flat surface and then pulled in four directions {↑, ↓, ←, →} (i.e., up, down, left, right) which results in five unique configurations of each piece of laundry. In other words, deliberate actions change the state of the world in

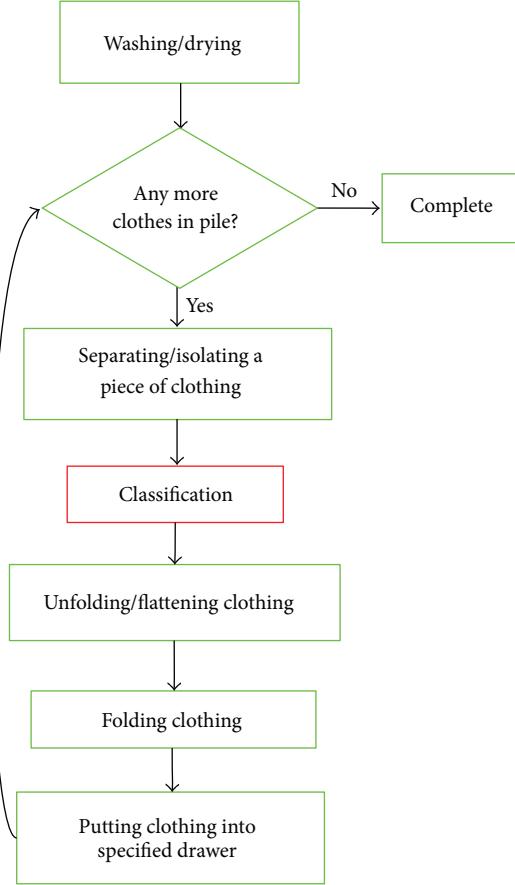


FIGURE 1: Overview of the laundry process, adapted from [10]. Green areas represent parts of the process that have already been explored in previous work, while the Red area represents the part of the process that is the focus of this paper.

a way that simplifies perception and consequently future interactions.

In this paper, we demonstrate the approach based on characteristics, which was introduced in [15], for automatically classifying and sorting laundry. We extend [15] by applying this approach to a larger database. This new database increases the number of midlevel characteristics as well as increasing the number of categories. This paper will display the results of the algorithm by increasing the complexity of the problem. Figure 2 and Table 1 illustrate examples of laundry items from several actual household hampers which form the data set that will be used in this paper. Our new approach is comprised of a multilayer classification strategy consisting of SVM [23] and bag-of-words model [24], similar to [25], that utilizes characteristics and selection masks to bridge together each calculated low-level feature to the correct category for each article of clothing.

## 2. Previous Work

There has been previous work in robot laundry which relates to the research in this paper. Kaneko and Kakikura [11] focused on a strategy to isolate and classify clothing. The work

in [11] is an extension of the authors' pioneering work in [10]. In these papers, the authors categorize what steps and tasks are needed for laundry (isolate, unfold, classify, fold, and put away clothes), as in Figure 1. The authors classify each object into one of three groups (shirt, pants, or towel) by deforming the article of clothing. The process of deforming the object aims to find hemlines as grasp points and the grasp points cannot be closer than a predefined threshold. Once the object is held at two grasp points, then the shape of the object is broken up into different regions and the values of the different regions are used to calculate feature values. Each of the feature values is compared against threshold values. If the features are above or below the thresholds, then the object is classified as a shirt, pant, or towel based on a decision tree. The results achieved a 90% classification rate.

Osawa et al. [12] use two manipulators to pick up/separate a piece of cloth, unfold, and classify it. The approach in [12] begins the process by picking up a piece of cloth by holding it with one manipulator. While the item is being held by the first manipulator, the second manipulator looks for the lowest point of the cloth as a second grasp point. The grasp spreads the article of clothing by the longest two corner points. The article of clothing now held by both robot arms is imaged by a camera and the image is converted into a binary image using background subtraction. This binary image is then compared with the binary template to classify the article of clothing. The approach is tested with eight different categories (male short sleeve shirt, women/child short sleeve shirt, long sleeve shirts, trousers, underwear shorts, bras, towels, and handkerchiefs). Fifty trials were conducted with clothing of the same type as the training data, achieving an average of 96% classification.

Kita et al. [6] extend their original work [4] of determining the state of clothing. The work in [6] provides a three-step process to successfully find a grasp point that the humanoid robot can grab: (1) clothing state estimation (previous work [4]); (2) calculation of the theoretically optimal position and orientation of the hand for the next action; and (3) modification of position and orientation, based on the robots limitations, so the action can be executed. The 3D model is used by calculating the orientation of the shirt model at the grasp point. The 3D model gives three vectors around the grasp point and the vectors are used in the calculation of the orientation. The authors tested their results of six scenarios and were successful on five. One scenario failed in predicting the correct clothing state, four scenarios were successful but had to use step (3) for completion, and the last scenario was successful without using step (3).

The results in [13] are another extension to Y. Kita and N. Kita's previous work [4]. The purpose is to accurately determine the shape/configuration of a piece of clothing based on the observed 3D data and the simulated 3D model. Each of the clothing states is compared against the 3D observed data based on the length of the clothing as a first step. Each of the shape states within a threshold value is selected to use in the next step of the process. Then, a correlation function is applied to the 2D plane of the observed data and simulated data to calculate which shape state model is closest to the observed data. The experimental results show that 22 out of 27 attempts were correctly classified as the correct shape state.

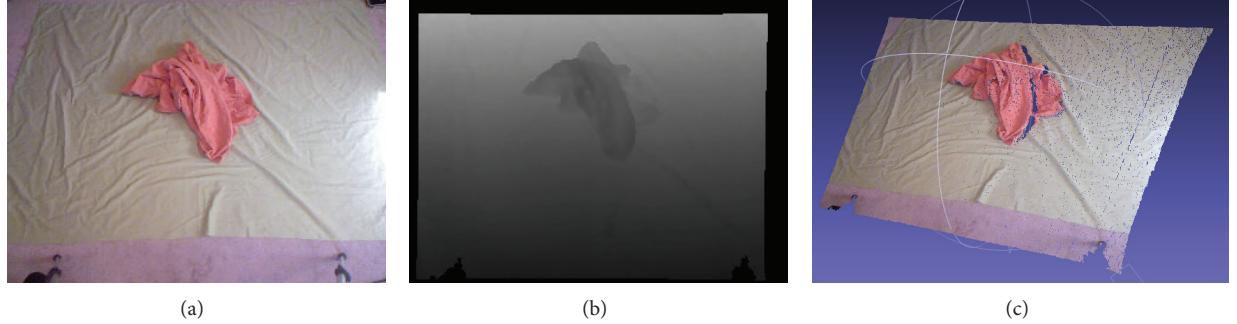


FIGURE 2: The proposed setting for classification. The system classifies the items automatically by comparing the color, texture, shape, and size of the object with those of a learned database. Sensory information is provided by a Kinect sensor. Data to be used in our approach. From left to right: color image, depth image, and a point cloud.

TABLE 1: Seven instances of clothing for each of the seven categories. From top row to bottom row: shirts, cloths, pants, shorts, dresses, socks, and jackets.



Cusumano-Towner et al. [14] present an extension of Osawa et al. [12] and Kita et al. [13] combined. The authors identified a clothing article by matching together silhouettes of the article to precomputed simulated articles of clothing. The silhouettes of each iteration were placed into a hidden markov model (HMM) to determine how well the silhouettes of the current article match against the simulated models of

all categories of clothing. The authors placed the article on a table by sliding the clothing along the edge of the table to maintain the configuration while it was hanging in the air. The authors grasped two locations on the cloth during each iteration. The planning algorithm that determines where the next set of grasp points are located was computed by using a directed grasp-ability graph that locates all possible grasp

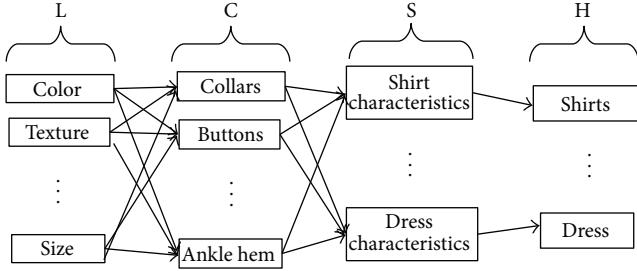


FIGURE 3: The L-C-S-H hierarchy with categories at the high level, binary vectors at the selection levels, attributes at the characteristic level, and features at the low-level.

points that can allow the system to reach. The authors tested their approach on seven articles of clothing of the end-to-end approach and were successful with 20 out of 30 trials.

Our work differs from the previous papers [6, 11–14] in that we work with a single manipulator and our results do not depend on hanging our clothing in the air by two grasp points. Our approach is independent of the type or how many manipulators that are used to collect the data. Our work also differs from that of [6, 13, 14] in that we do not require or use predefined shape models. We instead use novel characteristic features that describe each category of clothing. The characteristic features are not predefined but calculated through training the algorithm with a priori collected clothing data.

Our current work extends the problem of our previous work [15] by expanding the midlevel layer with more characteristics and incorporating more categories in the high layer. We expand the database of clothing that was used in our previous paper from 117 articles to over 200 articles of various size, color, and type. We then compare how well our algorithm improves over a baseline system of SVMs, one per category, to classify a pile of laundry into three, four, and seven groups (Tables 3 and 4).

### 3. Approach

**3.1. Overview.** We call the proposed algorithm L-C-S-H, for low-level-characteristics-selection mask-high level. Low-level refers to the features that are calculated from the images of each article (e.g., 2D shape, 2D color, 2D texture, 2D edges, and 3D shape). Characteristics refer to the attributes that are found on generic articles of clothing (e.g., pockets, collars, hems). Selection mask refers to a vector that describes what characteristics are most useful for each category (e.g., collars and buttons are most useful for classifying shirts), used as a filter. High level refers to the categories that we are attempting to classify (e.g., shirts, socks, dresses). Figure 3 illustrates the path of the L-C-S-H process. The L-C-S-H algorithm is described in detail in later sections.

**3.2. Clothing Database.** The database that we used consists of over 200 articles of clothing in over 1000 total configurations. The entire database is separated into seven categories and

more than three dozen subcategories. Each article of clothing is collected from real laundry hampers to better capture actual items encountered in the real world. Each one is laid flat on a table in a canonical position and dropped on the table in a crumpled position from four random grasp points for a total of five instances per article. Figure 1 illustrates seven instances of clothing for each of the seven categories that we used.

The database contains 2D color images, depth images, and 3D point clouds. For this paper, we will utilize the assortment of different types of items, such as shirts, cloths, pants, shorts, dresses, socks, and jackets, that might be encountered by a domestic robot involved in automating household laundry tasks. Figure 2 illustrates an example of a single color image, a single depth image, and a single point cloud for one article.

**3.3. Low-Level Features.** The L component of the approach uses the low-level features to estimate if the article does or does not have a particular characteristic. The low-level features that were used in this approach consist of color histogram (CH), histogram of line lengths (HLL), table point feature histogram (TPFH), boundary, scale-invariant feature transform (SIFT), and fast point feature histogram (FPFH). Each low-level feature will be described in detail in further subsections. The database that we used consisted of five instances of each article, see Section 3.2. In order to combine the low-level features of all five instances into a single value or histogram, we calculated each value by averaging each individual value along with its neighbors, in the case of the histogram. Equation (1) was the computation used to combine all five instances of each article

$$C_j = \frac{1}{N} \sum_{j=1}^N v_j + \frac{1}{2} (v_{j-1} + v_{j+1}), \quad (1)$$

where  $C_j$  is the combined score of the article,  $N$  is the number of instances of each article (we use  $N = 5$ ),  $v_j$  is the current value within the histogram along with  $v_{j-1}$  and  $v_{j+1}$  to be the immediate neighbors to the left and right of the current value, respectively.

For the part of the algorithm that converts from low-level to characteristics, we compared the low-level features, described in the following subsections, to the various characteristics. Since the characteristics were binary values, we used libSVM [23] to solve the two-class problem. We used each low-level feature to determine if the characteristic is in class 1 or 2. Class 1 contains positive instances and class 2 contains negative instances.

For an article of clothing, we capture an RGB image and a raw depth map, from which we get a 3D point cloud. We perform background subtraction on the RGB image to yield an image of only the object. The background subtraction is performed using graph-based segmentation [26], see Section 3.3.1. Once the object is isolated within the image, multiple features are calculated from the RGB image and the 3D point cloud. These features, which are discussed in later sections, capture 2D shape, 2D color, 2D texture, 2D edges, and 3D shape for both global and local regions of the object.

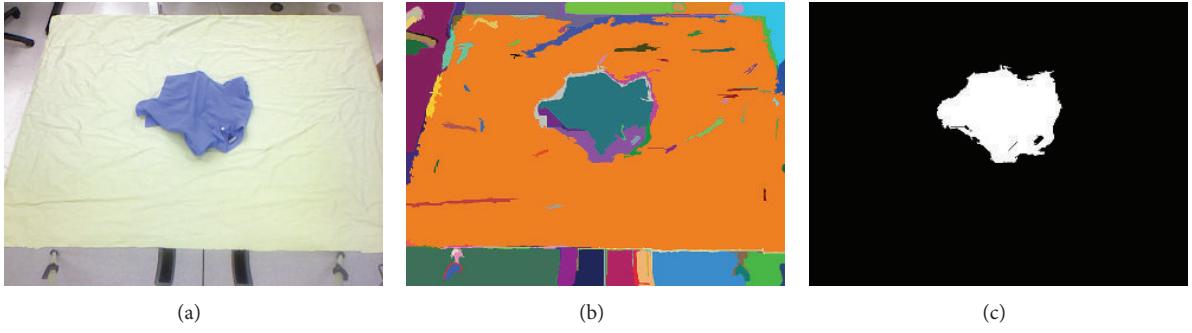


FIGURE 4: (a) An image taken by one of the overhead cameras in our setup. (b) The results of applying the graph-based segmentation algorithm to locate the foreground. (c) The binary image represents the location of the foreground object.

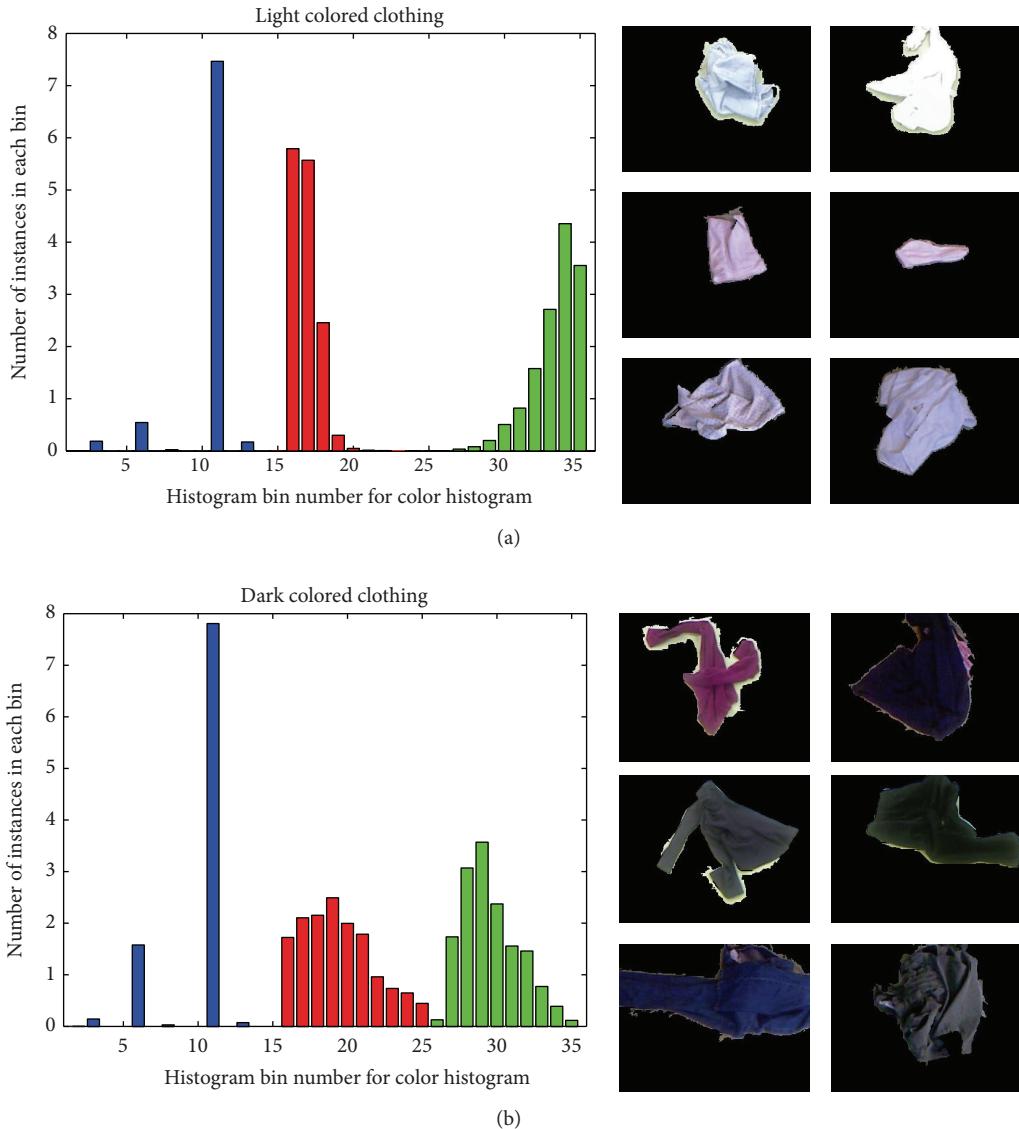


FIGURE 5: Resulting histograms of light (a) and dark (b) colored clothing. Each plot shows the histogram of the HSV color space, which consists of three parts: hue (blue), saturation (red), and value (green). For each category, the 6 examples used to compute the histogram are shown.

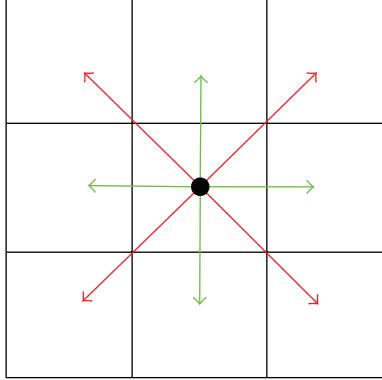


FIGURE 6: illustrative Example of diagonal movements and isothetic movements. The red arrows refer to diagonal movements and green arrows refer to isothetic movements.

**3.3.1. Graph-Based Segmentation.** We use Felzenswalb and Huttenlocher’s graph-based segmentation algorithm [26] because of its straightforward implementation, effective results, and efficient computation. This algorithm uses a variation of Kruskal’s minimum spanning tree algorithm to iteratively cluster pixels in decreasing order of their similarity in appearance. An adaptive estimate of the internal similarity of the clusters is used to determine whether to continue clustering. Figure 4 shows the results of graph-based segmentation on an example of  $640 \times 480$  RGB color image taken in our lab using the default value for the scale parameter ( $k = 500$ ). As can be seen, the segmentation provides a reasonable representation of the layout of the items in the pile. From this result, we determine the foreground regions as those that do not touch the boundary of the image.

### 3.4. Global Features

**3.4.1. Color Histogram (CH).** A CH is a representation of the distribution of the colors in a region of an image, derived by counting the number of pixels with a given set of color values. CH are chosen in this work because they are invariant to translation and rotation about the viewing axis, and for most objects they remain stable despite changes in viewing direction, scale, and 3D rotation. CH is used to distinguish, for example, between lights and darks, as well as denim. We compute the CH of each object in HSV color space. The HSV color space is computed by converting the RGB space to hue, saturation, and value. Equations (2)–(6) display the equations needed to compute the hue of a pixel. In the case of  $C = 0$ , the first bin in the hue color channel is incremented to keep track of all undefined instances. Equations (7)–(8) display the equations needed to compute the value and saturations of a pixel, respectively. We use 15 bins for hue color channel and 10 bins for the saturation and value color channels, leading to 35 total bins. The hue ranges from  $0 \rightarrow 360$  degrees; the saturation and the value both range from  $0.0 \rightarrow 1.0$ . Figure 5 illustrates the difference in an article of clothing with light colored and dark colored fabric

$$M = \max(R, G, B), \quad (2)$$

$$m = \min(R, G, B), \quad (3)$$

$$C = M - m, \quad (4)$$

$$H' = \begin{cases} \text{undefined}, & \text{if } C = 0, \\ \frac{G-B}{C} \bmod 6, & \text{if } M = R, \\ \frac{B-R}{C} + 2, & \text{if } M = G, \\ \frac{R-G}{C} + 4, & \text{if } M = B, \end{cases} \quad (5)$$

$$H = 60^\circ \times H', \quad (6)$$

$$V = \max(R, G, B), \quad (7)$$

$$S = \frac{C}{V}. \quad (8)$$

**3.4.2. Histogram of Line Lengths (HLL).** The histogram of line lengths (HLL) is a novel feature that we are introducing to help distinguish between stripes, patterns, plaid, and so forth. For this, we use the object image as before (after background subtraction) and compute the Canny edges [27], then erode with a  $3 \times 3$  structuring element of ones to remove effects of the object boundary. Then we compute the length of each Canny edge using Kimura et al.’s [28] method. The Kimura et al.’s method uses diagonal movement ( $N_d$ ) and isothetic movement ( $N_o$ ) to accurately calculate the length of a pixelated edge within an image, see (9). Diagonal movements ( $N_d$ ) refer to moving from the center pixel to one of the four corners within a  $3 \times 3$  window and isothetic movements ( $N_o$ ) refer to moving directly left, right, up, or down, see Figure 6. These pixel lengths are used to compute a histogram of 20 bins that range from 0 pixels to 1000 pixels, so each bin captures lengths within 50 pixels. Lengths greater than 1000 get mapped down to the last bin. Figure 7 illustrates the difference in an article of clothing with no texture, with stripes, and with plaid

$$L = \left[ N_d^2 + \left( N_d + \frac{N_o}{2} \right)^2 \right]^{1/2} + \frac{N_o}{2}. \quad (9)$$

**3.4.3. Table Point Feature Histogram (TPFH).** The TPFH consists of a 263-dimension array of float values that result from three 45-value subdivisions, that are calculated from extended fast point feature histograms (eFPFH), and 128-value subdivision for table angle information. This feature is a variant on the viewpoint feature histogram (VFH) [29]. The eFPFH values are calculated by taking the difference of the estimated normals of each point and the estimated normal of the objects centerpoint. The estimated normals of each point and the centerpoint are calculated by projecting them on the XY, YZ, and XZ plane. The differences between the two normals are labeled  $\beta$ ,  $\theta$ , and  $\phi$ . These three values, ranging from  $-90 \rightarrow +90$  degrees (4 degrees each bin), are placed within bins in the three 45-value histograms of the eFPFH. Figure 8 illustrates the graphical representation of the normals and how they are calculated. The 128-value table

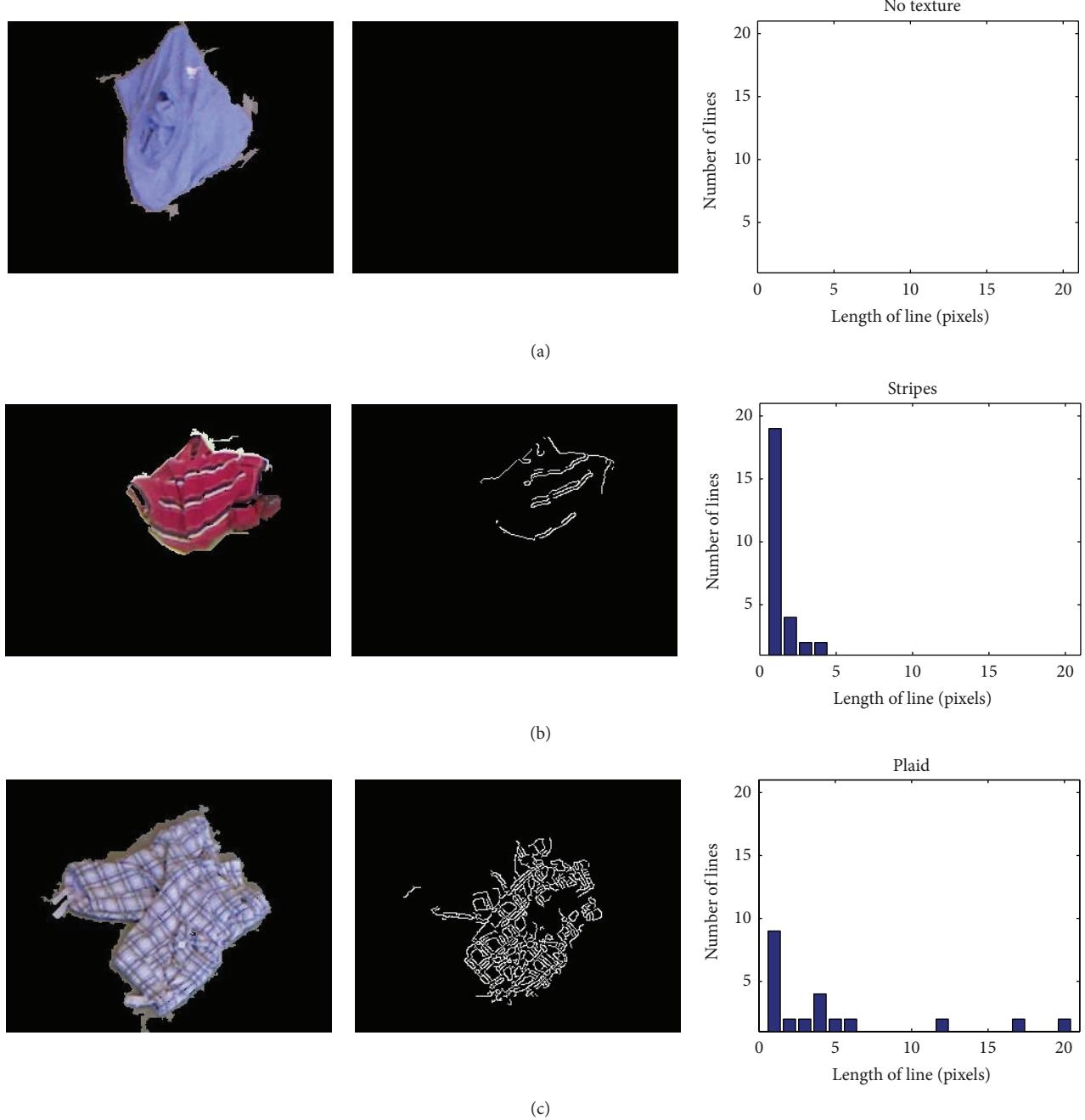


FIGURE 7: Resulting histograms of an article of clothing with no texture (a), with stripes (b), and with plaid (c). Each row contains the original image, the image after using Canny edge detector, and the histogram of line lengths.

histogram is computed by finding the angle,  $\alpha$ , between each normal vector and the translated central table vector for each point. The central table vector is translated to the same point as the normal that is currently being computed. Figure 8 illustrates the difference in direction for each vector. In TPFH, the eFPFH component is a translation, rotation, and scale invariant, while the table component is only a scale invariant, 3D representation of the local shape. Figure 9 illustrates an example of how the TPFH values are visually different in two separate categories.

**3.4.4. Boundary.** The boundary feature captures 2D shape information by storing the Euclidean distances from the centroid of each article to the boundary. First, the centroid of each binary image is calculated containing the object (after background subtraction). Then, starting at the angle of the major axis found by principle components analysis, 16 angles that range from 0 to 360 (i.e., 0 to 337.5) are calculated around the object. Angles are 0, 22.5, 45, 67.5, ..., 315, 337.5. For each angle, we measure the distance from the centroid to the furthest boundary pixel, see Figure 10.

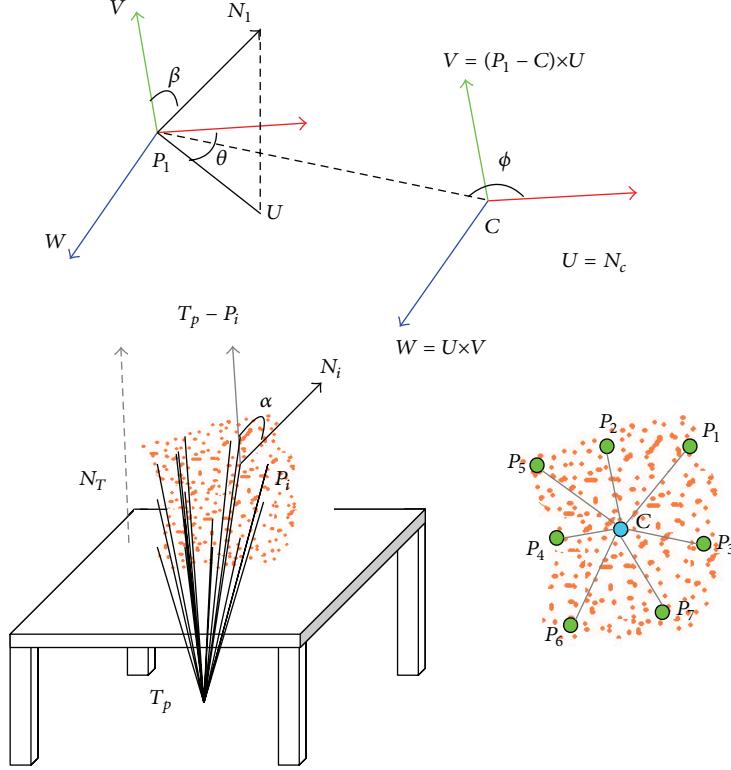


FIGURE 8: Graphical representation of the two components of TPFH. Top: eFPFH component with  $\beta$ ,  $\theta$ , and  $\phi$  shown.  $V$ ,  $W$ , and  $U$  represent the coordinates of point  $C$  (centroid), with respect to the normal of  $C$  and the position of  $C$  and  $P_1$ .  $N_1$  is the normal of point  $P_1$ . Bottom: Table component with  $\alpha$  shown.  $T_p$  is the centroid of the table, while  $N_T$  is the normal of the table.  $N_i$  is the normal of point  $P_i$  and  $P_1 \rightarrow P_7$  are arbitrary points in the cloud surrounding point  $C$ .

TABLE 2: List of 27 unique characteristics and the number of instances for each characteristic that were chosen to differentiate categories of clothing. The number of articles containing each characteristic is indicated in parentheses.

Collar (21)	Front pockets (39)	Back pockets (26)	Side pockets (8)	Elastic band (32)	Front zipper (28)
Top brackets (10)	Graphic pictures (47)	Graphic texts (7)	Belt loops (25)	Top buttons (30)	Shoulder hem (15)
Dark colored (70)	Colored (94)	White colored (38)	V-neck (7)	Striped (11)	Wrist hem (26)
Denim (13)	Plaid (5)	Patterns (18)	Round neck (73)	Bicep hem (59)	Shin hem (15)
Ankle hem (30)	Thigh hem (28)	Inseam (52)			

### 3.5. Local Features

**3.5.1. SIFT Descriptor.** The SIFT, scale invariant feature transform [30], descriptor is used to gather useful 2D local texture information. The SIFT descriptor locates points on the article (after background subtraction) that provide local extremum when convolved with a Gaussian function. These points are then used to calculate a histogram of gradients (HoG) from the neighboring pixels. The descriptor consists of a 128-value feature vector that is scale and rotation invariant. Figure 11 illustrates the SIFT feature points found on an article. The arrows represent the orientation and magnitude of the feature point. After all of the feature points are found on an article, the SIFT descriptors are placed within a bag-of-words model [24] to calculate 100 codewords. These codewords provide each article with a 100-element feature vector that represents the local texture information.

**3.5.2. FPFH.** The FPFH, fast point feature histogram [31], descriptor is used to gather local 3D shape information. The FPFH descriptor utilizes the 3D point cloud and background subtraction for each article and segments the article from the background of the point cloud. For each 3D point, a simple point feature histogram (SPFH) is calculated by taking the difference of the normals between the current point and its  $k$  neighboring points with a radius  $r$ . Figure 12 illustrates an example of a 3D point along with its neighbors. The radius is precomputed for each point cloud to best capture local shape information. Once all of the SPFHs are computed, the FPFH descriptor of each point is found by adding the SPFH of that point along with a weighted sum of the  $k$  neighbors, see (10). FPFH is a histogram of 33 values, that is, three sets of 11 bins for the three different orthogonal planes of XY, YZ, and XZ

$$\text{FPFH}(p) = \text{SPFH}(p) + \frac{1}{k} \sum_{i=1}^k w_i \text{SPFH}(i). \quad (10)$$

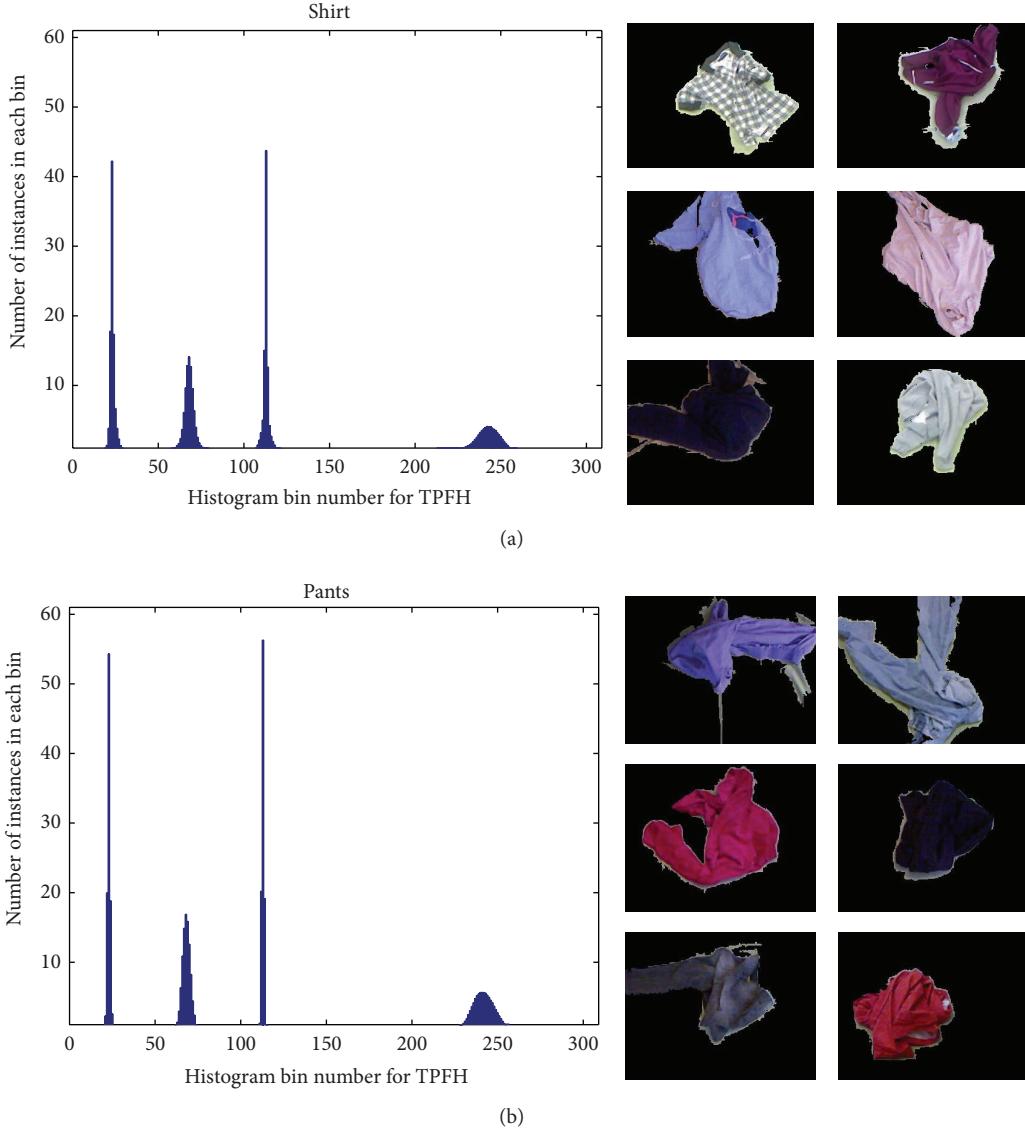


FIGURE 9: Example of the visual differences in global shape information between two different categories, namely, shirts (a) and pants (b). Each row contains the average histogram of the TPFH (left) and 6 examples of each category used to calculate the average (right). Shirts have a relatively lower number of instances in the first and third peak than pants. This example shows why the TPFH feature is useful in classifying clothing.

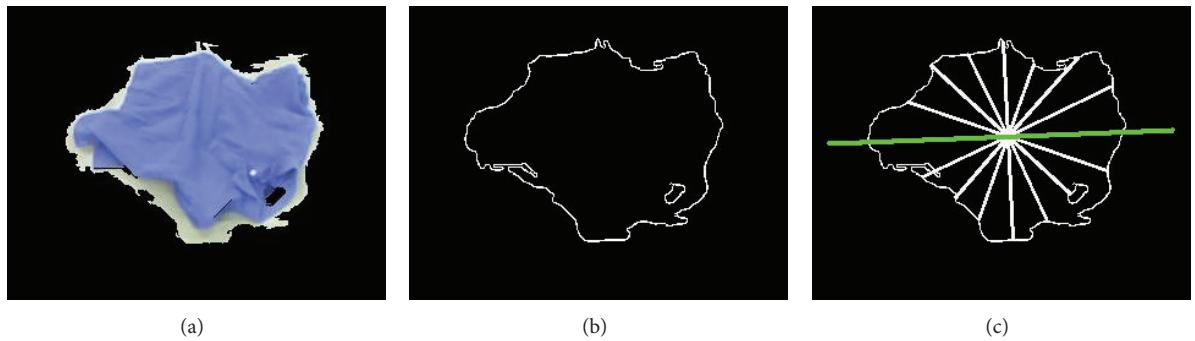


FIGURE 10: (a) Article segmented from background. (b) The binary boundary of the article. (c) The boundary of the article along with the lines connecting the centroid to the boundary, whose distances are used in the feature vector. The green line represents the major axis found by PCA.

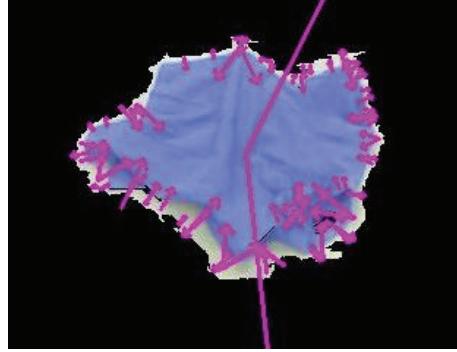


FIGURE 11: Resulting SIFT features overlayed on the image. The arrows represent the orientation and scale (magnitude) of each SIFT descriptor.

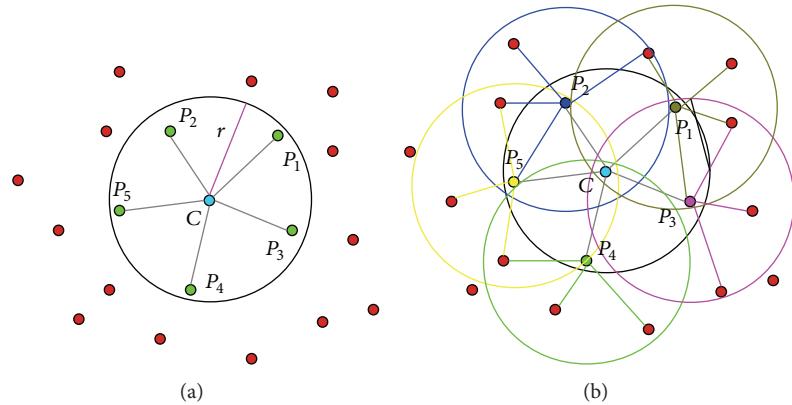


FIGURE 12: (a) Example of the SPFH of a single point. An area of radius  $r$  encircles 5 neighboring points for this example. (b) An example of the FPFH of the current point along with surrounding SPFHs used in the reweighting scheme. Each neighboring SPFH is color coded to represent the surrounding connections within the various regions.

TABLE 3: Results of baseline system in previous work.

	Shirt	Dress	Socks
Local features			
Shirt	<b>98.82</b>	1.18	0.00
Dress	100.00	<b>0.00</b>	0.00
Socks	0.00	13.64	<b>86.36</b>
Average TPR = <b>61.73%</b>			
Global features			
Shirt	<b>57.65</b>	2.35	40.00
Dress	60.00	<b>0.00</b>	40.00
Socks	36.36	18.18	<b>45.45</b>
Average TPR = <b>34.37%</b>			
Local and global features			
Shirt	<b>40.00</b>	0.00	60.00
Dress	40.00	<b>0.00</b>	60.00
Socks	18.18	27.27	<b>54.55</b>
Average TPR = <b>31.52%</b>			

TPR stands for true positive rate.

**3.5.3. Final Input Feature Vector.** Once the features are computed, the global features are concatenated to create a histogram of  $35 + 20 + 263 + 16 = 334$  values. For local features, SIFT and FPFH are calculated separately through bag-of-words to get two 100-element histograms of codewords.

Being concatenated, this yields 200 values for the local features. Then being concatenated with global features yields 534 values, which are then fed to the multiple one-versus-all SVMs.

### 3.6. Midlevel Layers

**3.6.1. Characteristics.** The C component of the approach uses the characteristics found on everyday clothing that is best suited for each category. We use a binary vector of 27 values that correspond to each characteristic used to learn what attributes are needed to separate the differences between shirts and dresses. The 27 characteristics that were chosen are shown in Table 2, along with the number of instances of each characteristic in the database. Figure 13 illustrates the breakdown of how much each characteristic is used per category. Light red blocks represent small percentages, dark purple blocks represent high percentages, and blank areas represent a percentage of zero. The color scheme changes from red to purple as the percentage increases.

**3.6.2. Selection Mask.** The S component of the approach uses the characteristics to determine which subset is best suited for each category. The selection masks are stored as a binary value; 0 or *FALSE*, means that the category does not need

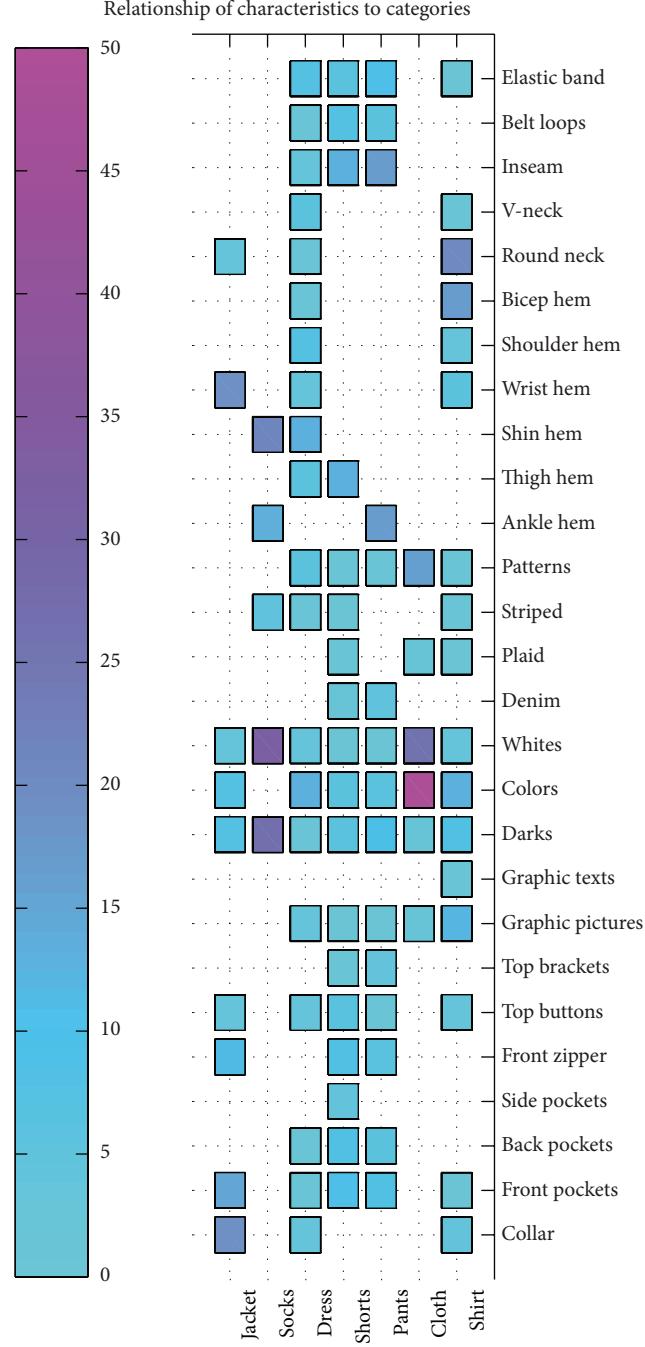


FIGURE 13: A graphical representation of the percentage of characteristics that are contained within each category. The color scheme changes from red to purple as the percentage increases. Light red colors represent small percentages. Dark purple colors represent high percentages. Blank areas represent a percentage of zero.

that particular characteristic, and 1 or *TRUE*, means that the category does need that particular characteristic.

We aimed to determine what characteristics have a higher importance and which have a lower importance for each category. Therefore, we zeroed one characteristic at a time and viewed the resulting percentage over the rest of the characteristics. The percentage was determined by comparing the characteristic vector for each article against mean vectors

for each category, which are calculated in Section 3.7. If the current percentage was higher or equal to the previously calculated percentage, then we zeroed that particular characteristic for the next iteration.

Importance values closer to 1.00 represent a positive type of instance and importance value scloser to 0.00 represent a negative type of instance. Importance values closer to 1.00 were chosen based on the fact that for one particular category,

TABLE 4: Results of baseline system using SVM to classify objects within seven categories.

	Shirt	Cloth	Pants	Shorts	Dress	Socks	Jacket
Local features							
Shirt	<b>98.82</b>	0.00	1.18	0.00	0.00	0.00	0.00
Cloth	63.33	<b>0.00</b>	20.00	10.00	0.00	6.67	0.00
Pants	100.00	0.00	<b>0.00</b>	0.00	0.00	0.00	0.00
Shorts	96.00	0.00	4.00	<b>0.00</b>	0.00	0.00	0.00
Dress	100.00	0.00	0.00	0.00	<b>0.00</b>	0.00	0.00
Socks	0.00	0.00	0.00	13.64	0.00	<b>86.36</b>	0.00
Jacket	100.00	0.00	0.00	0.00	0.00	0.00	<b>0.00</b>
Average TPR = <b>26.46%</b>							
Global features							
Shirt	<b>48.24</b>	0.00	1.18	10.59	0.00	40.00	0.00
Cloth	13.33	<b>50.00</b>	3.33	23.33	0.00	10.00	0.00
Pants	28.00	0.00	<b>0.00</b>	16.00	0.00	56.00	0.00
Shorts	44.00	0.00	4.00	<b>16.00</b>	0.00	36.00	0.00
Dress	40.00	0.00	0.00	20.00	<b>0.00</b>	40.00	0.00
Socks	18.18	9.09	0.00	27.27	0.00	<b>45.45</b>	0.00
Jacket	60.00	0.00	0.00	20.00	0.00	20.00	<b>0.00</b>
Average TPR = <b>22.81%</b>							
Local and global features							
Shirt	<b>1.18</b>	0.00	4.71	34.12	0.00	60.00	0.00
Cloth	0.00	<b>0.00</b>	13.33	43.33	0.00	43.33	0.00
Pants	0.00	0.00	<b>8.00</b>	32.00	0.00	60.00	0.00
Shorts	0.00	0.00	8.00	<b>32.00</b>	0.00	60.00	0.00
Dress	0.00	0.00	0.00	40.00	<b>0.00</b>	60.00	0.00
Socks	0.00	0.00	4.55	45.45	0.00	<b>50.00</b>	0.00
Jacket	0.00	0.00	0.00	40.00	0.00	60.00	<b>0.00</b>
Average TPR = <b>13.03%</b>							

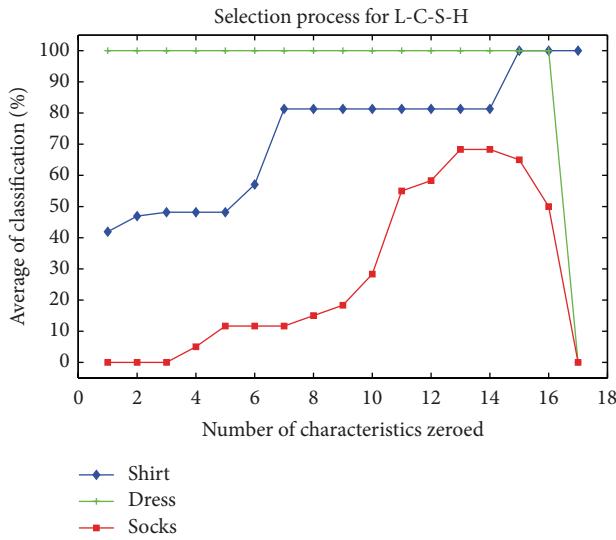


FIGURE 14: Characteristic selection process for L-C-S-H: results of removing a single redundant characteristic at each iteration. Starting from removing no characteristics and finishing with removing all characteristics.

TABLE 5: Results of L-C-H proposed system in previous work.

	Shirt	Dress	Socks
Local features			
Shirt	<b>41.18</b>	58.82	0.00
Dress	40.00	<b>60.00</b>	0.00
Socks	0.00	90.91	<b>9.09</b>
Average TPR = <b>36.76%</b>			
Global features			
Shirt	<b>15.29</b>	84.71	0.00
Dress	40.00	<b>60.00</b>	0.00
Socks	13.64	86.36	<b>0.00</b>
Average TPR = <b>25.10%</b>			
Local and global features			
Shirt	<b>21.18</b>	78.82	0.00
Dress	20.00	<b>80.00</b>	0.00
Socks	36.36	63.64	<b>0.00</b>
Average TPR = <b>33.73%</b>			

there is a likelihood that the category will contain a characteristic, more often, over the other categories. Importance

TABLE 6: Results of L-C-H proposed system to classify clothing using seven categories.

	Shirt	Cloth	Pants	Shorts	Dress	Socks	Jacket
Local features							
Shirt	<b>41.18</b>	0.00	0.00	0.00	58.82	0.00	0.00
Cloth	23.33	<b>0.00</b>	6.67	0.00	70.00	0.00	0.00
Pants	44.00	0.00	<b>0.00</b>	0.00	56.00	0.00	0.00
Shorts	36.00	0.00	0.00	<b>0.00</b>	64.00	0.00	0.00
Dress	40.00	0.00	0.00	0.00	<b>60.00</b>	0.00	0.00
Socks	0.00	0.00	54.55	13.64	31.82	<b>0.00</b>	0.00
Jacket	20.00	0.00	0.00	0.00	80.00	0.00	<b>0.00</b>
Average TPR = <b>14.45%</b>							
Global features							
Shirt	<b>14.12</b>	0.00	4.71	0.00	80.00	0.00	1.18
Cloth	16.67	<b>0.00</b>	6.67	13.33	63.33	0.00	0.00
Pants	20.00	0.00	<b>4.00</b>	0.00	76.00	0.00	0.00
Shorts	28.00	0.00	4.00	<b>4.00</b>	64.00	0.00	0.00
Dress	40.00	0.00	0.00	0.00	<b>60.00</b>	0.00	0.00
Socks	13.64	0.00	0.00	0.00	86.36	<b>0.00</b>	0.00
Jacket	20.00	0.00	0.00	0.00	80.00	0.00	<b>0.00</b>
Average TPR = <b>11.73%</b>							
Local and global features							
Shirt	<b>21.18</b>	0.00	0.00	0.00	78.82	0.00	0.00
Cloth	10.00	<b>0.00</b>	0.00	0.00	90.00	0.00	0.00
Pants	24.00	0.00	<b>0.00</b>	0.00	76.00	0.00	0.00
Shorts	28.00	0.00	0.00	<b>0.00</b>	72.00	0.00	0.00
Dress	20.00	0.00	0.00	0.00	<b>80.00</b>	0.00	0.00
Socks	36.36	0.00	0.00	4.55	59.09	<b>0.00</b>	0.00
Jacket	20.00	0.00	0.00	0.00	80.00	0.00	<b>0.00</b>
Average TPR = <b>14.45%</b>							

TABLE 7: Results of L-C-S-H proposed system in previous work.

	Shirt	Dress	Socks
Local features			
Shirt	<b>95.80</b>	4.20	0.00
Dress	13.32	<b>86.65</b>	0.02
Socks	0.60	11.67	<b>87.72</b>
Average TPR = <b>90.06%</b>			
Global features			
Shirt	<b>90.41</b>	9.51	0.08
Dress	2.80	<b>96.95</b>	0.25
Socks	30.45	69.55	<b>0.00</b>
Average TPR = <b>62.45%</b>			
Local and global features			
Shirt	<b>58.64</b>	40.64	0.72
Dress	4.32	<b>95.67</b>	0.02
Socks	37.10	44.66	<b>18.24</b>
Average TPR = <b>57.52%</b>			

values closer to 0.00 were chosen based on the fact that for one particular category, there is a likelihood that the category will not contain a characteristic while the other categories

will contain that characteristic. Positive importance values can be converted to +1 and negative importance values can be converted to -1 to better understand how each characteristic contributes to that one category.

The next iteration repeated the same process of removing one characteristic at a time along with having the previously chosen characteristics removed. After permanently removing several features, the resulting classification percentage began to increase to a peak and after the peak began to decrease when we removed the remaining characteristics. Next, we then created a binary vector for each category that contained a 1 or *TRUE* if the characteristic has a higher importance and a 0 or *FALSE* if the characteristic has a lower importance. Algorithm 1 steps through the process of C-S.

**3.7. High-Level Categories.** The H component of the approach uses the characteristic vectors that correspond to the articles within the training set to average the binary vectors of each category together to create a mean vector, as a descriptor, for each category. In other words, all shirts have averaged their characteristic vectors together, all dresses have averaged their characteristic vectors, and so on to create seven unique mean vectors, one for each category. Then the selection mask

TABLE 8: Results of L-C-S-H proposed system to classify objects.

	Shirt	Cloth	Pants	Shorts	Dress	Socks	Jacket
Local Features							
Shirt	<b>95.80</b>	0.00	0.00	0.00	4.20	0.00	0.00
Cloth	28.08	<b>0.00</b>	0.00	0.00	69.45	2.48	0.00
Pants	53.33	0.00	<b>0.00</b>	0.00	46.67	0.00	0.00
Shorts	47.53	0.00	0.00	<b>0.00</b>	52.47	0.00	0.00
Dress	13.32	0.00	0.00	0.00	<b>86.65</b>	0.02	0.00
Socks	0.60	0.01	0.00	0.00	11.67	<b>87.72</b>	0.00
Jacket	33.81	0.00	0.00	0.00	66.19	0.00	<b>0.00</b>
Average TPR = <b>38.60%</b>							
Global Features							
Shirt	<b>90.41</b>	0.00	0.00	0.00	9.51	0.08	0.00
Cloth	23.04	<b>0.00</b>	0.00	0.00	76.90	0.06	0.00
Pants	37.04	0.00	<b>0.00</b>	0.00	62.94	0.02	0.00
Shorts	36.93	0.00	0.00	<b>0.00</b>	63.00	0.07	0.00
Dress	2.80	0.00	0.00	0.00	<b>96.95</b>	0.25	0.00
Socks	30.45	0.00	0.00	0.00	69.55	<b>0.00</b>	0.00
Jacket	23.36	0.00	0.00	0.00	76.64	0.00	<b>0.00</b>
Average TPR = <b>26.77%</b>							
Local and Global Features							
Shirt	<b>65.79</b>	0.00	0.00	0.89	31.49	0.04	1.79
Cloth	18.67	<b>0.00</b>	3.72	4.43	67.42	0.01	5.75
Pants	36.00	0.00	<b>1.93</b>	0.52	45.53	0.00	16.02
Shorts	32.94	0.00	1.72	<b>2.24</b>	45.62	0.05	17.43
Dress	5.48	0.00	0.00	0.07	<b>93.80</b>	0.00	0.65
Socks	37.54	0.00	1.26	3.17	50.06	<b>0.00</b>	7.97
Jacket	17.39	0.00	0.00	0.03	50.37	0.02	<b>32.20</b>
Average TPR = <b>27.99%</b>							

```

Input: Characteristic vector for  $w$ th category ( $CC_w$ )
Output: Selection mask for  $w$ th category ( $SM_w$ )
 $SM_w = 2^{26} - 1;$ 
 $RM_w = 2^{27} - 1;$ 
foreach Article $_i \in$  Category $_w$  do
     $NC_w = (CC_w \times SM_w) \cap RM;$ 
    If Result( $NC_w$ )  $\geq$  Result( $CC_w$ ) then
         $RM_w = RM_w \cap 2^{27} - 2^i - 1;$ 
    end
     $SM_w = SM_w \gg 1;$ 
end

```

ALGORITHM 1: The process of C-S to determine the final selection mask for each category.  $CC_w$ ,  $NC_w$ ,  $SM_w$ , and  $RM_w$  are binary vectors that represent if a characteristic is used (1) or removed (0), out of the 27 listed in Table 2, for each category.  $\gg$  stands for a bit shift in the binary vector. Result(-) calculates the resulting classification percentage after running the new selection mask on the training set.

is multiplied by the characteristic vector to zero out any characteristics with a negative importance.

#### 4. Experimental Results

The proposed approach was applied to a laundry scenario to test its ability to perform practical classification. In each experiment, the entire database consisted of 85 shirts, 30

cloths, 25 pants, 25 shorts, 10 dresses, 22 socks, and 5 jackets. The database was labeled using a supervised learning approach so that the corresponding category of each image was known. We provide an approach that illustrates the use of having midlevel characteristics (attributes) in grouping clothing categories that are difficult to classify. First, we consider the baseline approach that uses SVMs to directly classify each category without the use of midlevel characteristics. Then, we

TABLE 9: Overview of results for varying numbers of categories: (A) 3, (B) 4, and (C) 7. Group (A) consists of shirts, socks, and dresses. Group (B) consists of group (A) plus cloths. Group (C) consists of group (B) plus pants, shorts, and jackets.

Number of categories	Local	Global	Local + Global
Baseline			
3	61.73	34.37	31.52
4	46.3	38.28	19.22
7	26.46	22.81	13.03
L-C-S-H			
3	90.06	62.45	57.52
4	67.54	46.84	43.13
7	38.60	26.77	27.99
Increase from Baseline to L-C-S-H			
3	+28.33	+28.08	+26.00
4	+21.24	+8.56	+23.91
7	+12.14	+3.96	+14.96

introduce characteristics as an alternate mapping from low-level features to high-level categories.

**4.1. Baseline Approach without Midlevel Layers.** This experiment demonstrates the baseline approach using the low-level features to classify each article of clothing correctly with support vector machines (SVM). We consider various aspects of the SVM approach to better understand how the articles are categorized. This experiment uses local, global, and a combined set of low-level features as input for each SVM that is used to classify for a particular article. Since there are seven categories to choose from, then seven SVMs are trained in order to provide a probability value for each input vector from the test set. For this type of multiclass classification, we chose the one-versus-all (OVA) approach for each SVM, which uses a single SVM to compare each class with all other classes combined. Previous researchers [32–35] suggest that for sparse data sets, the OVA approach provides better results over the max-wins-voting (MWV) approach, which uses a single SVM for each pair of classes resulting in a voting scheme.

Each experiment was conducted using the 5-fold cross validation (CV) [36] to completely test this stage of the algorithm. We used 80% of each category for training and 20% for testing, with each train and test set being mutually exclusive for each validation. To better describe the separation, we used 80% of the database that had a 1 or *TRUE* for that category 80% of the database that had a 0 or *FALSE* for that category to make the training set, and the rest of the database went to the test set. The resulting value from each SVM ranges from  $-1 \rightarrow +1$ , with a deciding threshold of 0.

**4.2. Testing Approach with Characteristics (L-C-H).** For our experiment with L-C-H, we used 5-fold CV to test this stage of the algorithm, as in the previous experiments (Tables 5–8). The goal of this experiment is to determine the increase of overall performance with the addition of characteristics. The

training set still consists of the ground truth characteristics to compute the mean vectors for each category.

**4.3. L-C-H versus L-C-S-H.** For our experiment with L-C-S-H, we used 5-fold CV to test this stage of the algorithm, as in the previous experiments. The goal of this experiment is to determine the increase of overall performance with the addition of a selection mask between the characteristics and the high-level categories. The training set still consists of the ground truth characteristics to compute the mean vectors for each category. Each category demonstrates an increase in percentage until a threshold has been reached. Each threshold varies based on the low-level features used and what category is being tested. At that time, the percentage then eventually decreases to 0% after reaching the threshold.

Figure 14 illustrates the resulting TPR (true positive rate) as the most negatively important characteristic removed at each iteration, using local and global features. Each category demonstrates an increase in percentage until a threshold has been reached. This threshold is decided to coincide with the least amount of remaining characteristics that produce the highest TPR. At that time, the percentage then eventually decreases to 0% when the rest of the characteristics are zeroed out. So, zeroing out a subset of negatively important characteristics improves the overall TPR. Characteristic values for the mean category vector in the training set are the only values that are zeroed out. The training vectors go through this process to better describe how each category is characterized, while the testing vectors are not changed and compared to how close they are to each mean vector.

## 5. Conclusion

We have proposed a novel method to classify clothing (L-M-H, more specifically L-S-C-H) in which midlevel layers (i.e., physical characteristics and selection masks) are used to classify categories of clothing. The proposed system uses a database of 2D color images and 3D point clouds of clothing captured from a Kinect sensor. The paper is focused on determining how well this new approach improves classification over a baseline system. The database of clothing, that we use in our experiments, contains articles of clothing whose configurations are difficult to determine their corresponding category.

The overall improvement of this new approach illustrates the critical importance of middle-level information. The addition of middle-level features, termed characteristics and selection masks for this problem, enabled the classification process to improve, on average, by +27.47% for three categories, +17.90% for four categories and +10.35% for seven categories, see Table 9. Increasing the number of categories, increases the number of characteristics for the algorithm to classify. With the increase in complexity, the average true positive rate decreases due to the difficulty of the problem.

Using this approach could improve other types of classification/identification problems such as object recognition, object identification, person recognition. The use of middle-level features could range from one or more layers. This

approach does not have to be limited to using only two layers of features (e.g., characteristics and selection masks). The notion of adding more features in between the low and high level may increase the percentage rate. Layers of filters, that could be used to distinguish between categories, could include separating adult clothing from child clothing or male clothing from female clothing.

Future extensions of this approach include classification of subcategories (e.g., types of shirts, types of dresses), age, gender, and the season of each article of clothing. The experiments are used on a subset of characteristics that are useful for each category. Perhaps the amount of characteristics that are used in the midlevel layer correspond to the resulting classification percentages. This approach could be applied to separating clothing into three groups of darks, colors, and whites before the laundry is placed into the washing machine. This novel approach can apply to grouping and labeling other rigid and nonrigid items beyond clothing.

## Acknowledgment

This research was supported by the US National Science Foundation under Grants IIS-1017007 and IIS-0904116.

## References

- [1] S. Hata, T. Hiroyasu, J. Hayash, H. Hojoh, and T. Hamada, "Flexible handling robot system for cloth," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '09)*, pp. 49–54, August 2009.
- [2] Y. Yoshida, J. Hayashi, S. Hata, H. Hojoh, and T. Hamada, "Status estimation of cloth handling robot using force sensor," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '09)*, pp. 339–343, July 2009.
- [3] K. Salleh, H. Seki, Y. Kamiya, and M. Hikizu, "Tracing manipulation in clothes spreading by robot arms," *Journal of Robotics and Mechatronics*, vol. 18, no. 5, pp. 564–571, 2006.
- [4] Y. Kita and N. Kita, "A model-driven method of estimating the state of clothes for manipulating it," in *Proceedings of the 6th Workshop on Applications of Computer Vision*, pp. 63–69, 2002.
- [5] Y. Kita, F. Saito, and N. Kita, "A deformable model driven visual method for handling clothes," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3889–3895, May 2004.
- [6] Y. Kita, T. Ueshiba, E. Neo, and N. Kita, "A method for handling a specific part of clothing by dual arms," in *Proceedings of the Conference on Intelligent Robots and Systems (IROS '09)*, pp. 3403–3408, 2009.
- [7] P. Gibbons, P. Culverhouse, and G. Bugmann, "Visual identification of grasp locations on clothing for a personal robot," in *Proceedings of the Conference Towards Autonomous Robotics Systems (TAROS '09)*, pp. 78–81, August 2009.
- [8] H. Kobayashi, S. Hata, H. Hojoh, T. Hamada, and H. Kawai, "A study on handling system for cloth using 3-D vision sensor," in *Proceedings of the 34th Annual Conference of the IEEE Industrial Electronics Society (IECON '08)*, pp. 3403–3408, November 2008.
- [9] B. Willimon, S. Birchfield, and I. Walker, "Classification of clothing using interactive perception," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 1862–1868, 2011.
- [10] M. Kaneko and M. Kakikura, "Planning strategy for unfolding task of clothes—isolation of clothes from washed mass," in *Proceedings of the 13th annual conference of the Robotics Society of Japan (RSJ '96)*, pp. 455–456, 1996.
- [11] M. Kaneko and M. Kakikura, "Planning strategy for putting away laundry—isolating and unfolding task," in *Proceedings of the IEEE International Symposium on Assembly and Task Planning (ISATP '01)*, pp. 429–434, May 2001.
- [12] F. Osawa, H. Seki, and Y. Kamiya, "Unfolding of massive laundry and classification types by dual manipulator," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 11, no. 5, pp. 457–463.
- [13] Y. Kita, T. Ueshiba, E. S. Neo, and N. Kita, "Clothes state recognition using 3d observed data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 1220–1225, May 2009.
- [14] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *Proceedings of the International Conference on Robotics and Automation*, May 2011.
- [15] B. Willimon, I. Walker, and S. Birchfield, "A new approach to clothing classification using mid-level layers," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, 2013.
- [16] B. Willimon, S. Birchfield, and I. Walker, "Model for unfolding laundry using interactive perception," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, 2011.
- [17] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Parametrized shape models for clothing," in *Proceedings of the International Conference on Robotics and Automation*, pp. 4861–4868, May 2011.
- [18] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 272–277, May 2008.
- [19] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *Proceedings of the International Conference on Robotics and Automation (ICRA '09)*, pp. 1377–1382, 2009.
- [20] P. Fitzpatrick, "First contact: an active vision approach to segmentation," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS '03)*, pp. 2161–2166, 2003.
- [21] B. Willimon, S. Birchfield, and I. Walker, "Rigid and non-rigid classification using interactive perception," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 1728–1733, October 2010.
- [22] R. B. Willimon, *Interactive perception for cluttered environments* [M.S. thesis], Clemson University, 2009.
- [23] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001.
- [24] H. M. Wallach, "Topic modeling: beyond bag-of-words," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 977–984, June 2006.
- [25] A. Ramisa, G. Alenyá, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *Proceedings of the International Conference on Robotics and Automation*, pp. 1703–1708, 2012.

- [26] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [27] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [28] K. Kimura, S. Kikuchi, and S. Yamasaki, "Accurate root length measurement by image analysis," *Plant and Soil*, vol. 216, no. 1, pp. 117–127, 1999.
- [29] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 2155–2162, October 2010.
- [30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [31] R. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, Intelligent Autonomous Systems IAS, May 2009.
- [32] A. Gidudu, G. Hulley, and T. Marwala, "Image classification using SVMs: one-against-one vs one-against-all," in *Proceedings of the 28th Asian Conference on Remote Sensing*, 2007.
- [33] J. Milgram, M. Cheriet, and R. Sabourin, "One against one or one against all: which one is better for handwriting recognition with SVMs?" in *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition*, October 2006.
- [34] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.
- [35] K. bo Duan and S. S. Keerthi, "Which is the best multiclass SVM method? an empirical study," in *Proceedings of the 6th International Workshop on Multiple Classifier Systems*, pp. 278–285, 2005.
- [36] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall, London, UK, 1982.

