

A New Approach to Clothing Classification using Mid-Level Layers

Bryan Willimon, Ian Walker, and Stan Birchfield
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
{rwillim,iwalker,stb}@clemson.edu

Abstract— We present a novel approach for classifying items from a pile of laundry. The classification procedure exploits color, texture, shape, and edge information from 2D and 3D local and global information for each article of clothing using a Kinect sensor. The key contribution of this paper is a novel method of classifying clothing which we term L-M-H, more specifically L-C-S-H using characteristics(C), selection masks(S) layers and produces “local” solutions to solve the global classification problem. Experiments demonstrate the ability of the system to efficiently classify and label into one of three categories (shirts, socks, or dresses). These results show that, on average, the classification rates, using this new approach with mid-level layers, achieve a true positive rate of 90%.

I. INTRODUCTION

The general clothing classification problem is for a robotic system to properly and accurately sort a pile of laundry into specified, predefined groups (e.g., for loading the laundry into the washer, or after the drying process has completed). Laundry can be classified / sorted by category, age, gender, color (i.e. whites, colors, darks), season of use, or by individual. The specific problem addressed in this paper is classifying each article of clothing into a specified category (e.g. shirts, socks, dresses) using physical characteristics and selection masks to assist the process.

Other areas, along with classification, make up the process of “doing the laundry”. Figure 1 gives a flow chart of the various areas like separating/isolating [16] [28] [9] [10] [21], classifying [15] [28] [10] [21] [14] [4], unfolding / flattening [29] [4], and folding [20] clothes.

In contrast to previous work, our method operates on an unorganized, unflattened piece of laundry for the purpose of sorting each individual article. The database of clothing that we use in this paper consists of individual articles placed on a table by a single robotic manipulator. Previous clothing classification papers [15] [10] [21] [14] [4] used the ability of dual manipulators to freely hang each piece of laundry at two calculated grasp points to fully reveal the overall shape and outline of each article of clothing.

Our approach relies upon a combination of characteristics, a histogram for each article, and low-level image calculations in order to accomplish these objectives. 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

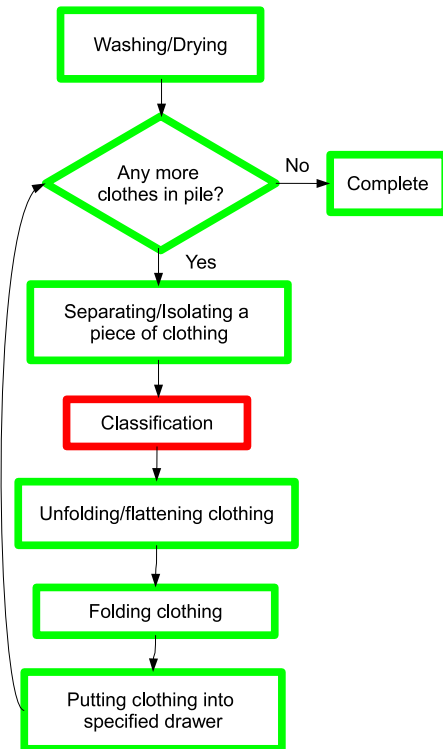


Fig. 1. Overview of the laundry process, adapted from [9]. GREEN areas represent parts of the process that have already been explored in previous work, while the RED area represents the part of the process that is the focus of this paper.

not obtainable through passive sensing alone [11] [12] [7] [27] [30]. The clothing that we use in this paper has been actively interacted with in order to gain more information about each article. In other words, deliberate actions change the state of the world in a way that simplifies perception and consequently future interactions.

In this paper, we present a new approach based on characteristics, which was inspired by [17], for automatically classifying and sorting laundry. The focus of this paper is to demonstrate that the framework of a multi-layer classification strategy will improve overall results. Each layer of the approach can be modified to include any type of learning strategy (e.g. SVM, k-NN, neural networks, Bayesian classifier, etc). In this paper, we limit our approach to one learning strategy since this is not our focus. Our new approach is comprised of a multi-layer classification strategy consisting

of SVM [3] and bag-of-words model [26], similar to [22], that utilizes characteristics and selection masks to bridge together each calculated low level feature to the correct category for each article of clothing.

II. APPROACH

A. Overview

The algorithm that we propose is acronymed **L-C-S-H** (**L**ow Level - **C**haracteristics - **S**election Mask - **H**igh Level). Low level refers to the features that are calculated from the images of each article using a Kinect sensor (e.g. 2D shape, 2D color, 2D texture, 2D edges, and 3D shape). Characteristics refers to the attributes that are found on generic articles of clothing (e.g. pockets, collars, hems, etc). 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 2 illustrates the path of the **L-C-S-H** process. The **L-C-S-H** algorithm is described in detail in later sections.

B. Clothing database

The database, that we used, consists of over 117 articles of clothing in over 585 total configurations. The entire database is separated into three categories and more than a 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 I illustrates seven examples of clothing for each of the three categories that we used.

The database contains 2D color images, depth images, and 3D point clouds that are captured using a Kinect sensor. For this paper, we will utilize the assortment of different types of items, such as shirts, socks, and dresses that might be encountered by a domestic robot involved in automating household laundry tasks. The articles of laundry are captured in this manner to test the possibility of developing a system capable to classifying clothes lying on the floor.

C. 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 low level features that are chosen for each article of clothing are used to determine / classify characteristics, shown in Table II. Nine out of the 17 characteristics can be calculated from shape information, seven out of 17 can be calculated from color information, five out of 17 can be calculated from texture information, and ten out of 17 can be calculated

from edge information. The database that we used consisted of five instances of each article, see section II-B. 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.

$$C_i = \frac{1}{N} \sum_{i=1}^N v_i + \frac{1}{2}(v_{i-1} + v_{i+1}) \quad (1)$$

where C_i is the combined score of the article, N is the number of instances of each article (we use $N = 5$), v_i is the current value within the histogram along with v_{i-1} and v_{i+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 [3] 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 results and class 2 contains negative results.

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 [6], see section II-C.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.

1) *Graph-based segmentation*: We use Felzenswalb and Huttenlocher’s graph-based segmentation algorithm [6] 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. From this result, we determine the foreground regions as those that do not touch the boundary of the image.

D. Global features

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

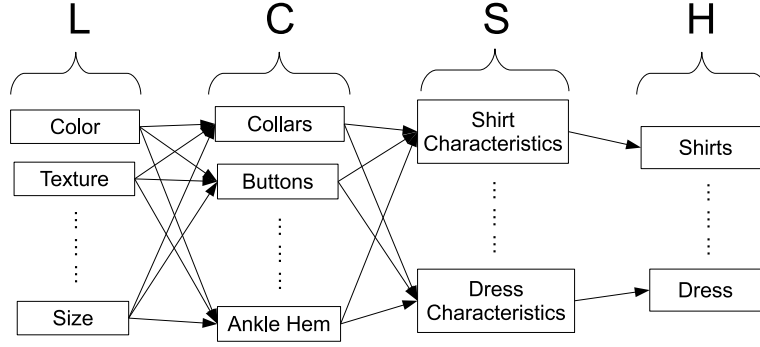


Fig. 2. The **L-C-S-H** heirarchy with categories at the High level, binary vectors at the Selection levels, attributes at the Characteristic level, and features at the Low level.

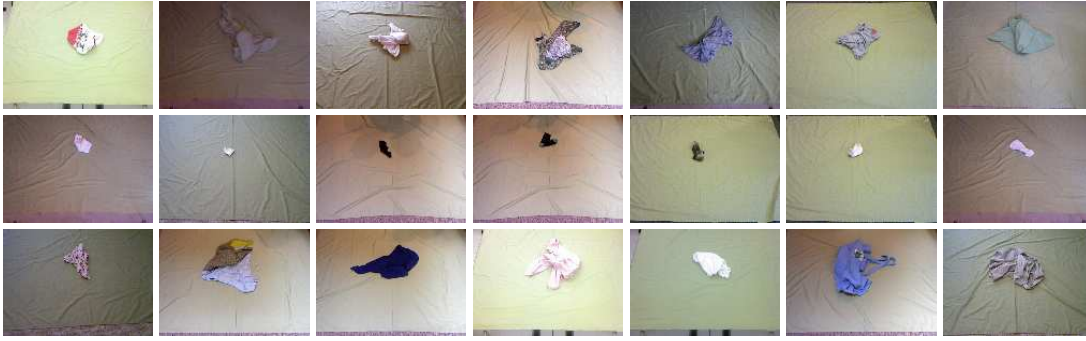


TABLE I

SEVEN EXAMPLES OF CLOTHING FOR EACH OF THE THREE CATEGORIES. FROM TOP ROW TO BOTTOM ROW: SHIRTS, SOCKS, AND DRESSES.

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 values. 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 $0.0 \rightarrow 1.0$.

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

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

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

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

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

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

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

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, etc. For this, we use the object image as before (after background subtraction) and compute the Canny edges [2], then erode with a 3×3 structuring element of ones to remove effects of the object boundary. Then we compute the length of each Canny edge using Kimura et al. [13] method. The Kimura et al. method uses diagonal movement (N_d) and isothetic movement (N_o) to accurately calculate the length of a pixelated edge within an image, see equation (9). Diagonal movements (N_d) refer to moving from the center pixel to one of the four corners within a 3×3 window and isothetic movements (N_o) refer to moving directly left, right, up, or down, see Figure 3. 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.

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

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) [25].

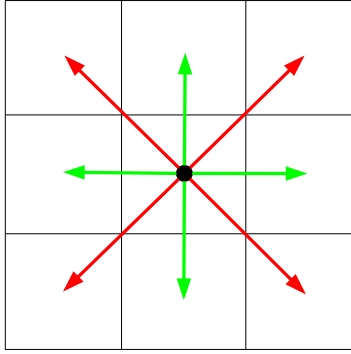


Fig. 3. Example illustration of diagonal movements and isothetic movements. The red arrows refer to diagonal movements and green arrows refer to isothetic movements.

The eFPFH values are calculated by taking the difference of the estimated normals of each point and the estimated normal of the objects center point. The estimated normals of each point and the center point are calculated by projecting them on the XY, YZ , and XZ plane. The differences between the two normals are labeled β, θ , and ϕ . 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 4 illustrates the graphical representation of the normals and how they are calculated. The 128 value table histogram is computed by finding the angle, α , 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 4 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.

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 5.

E. Local features

1) *SIFT descriptor*: The SIFT, Scale Invariant Feature Transform [18], 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

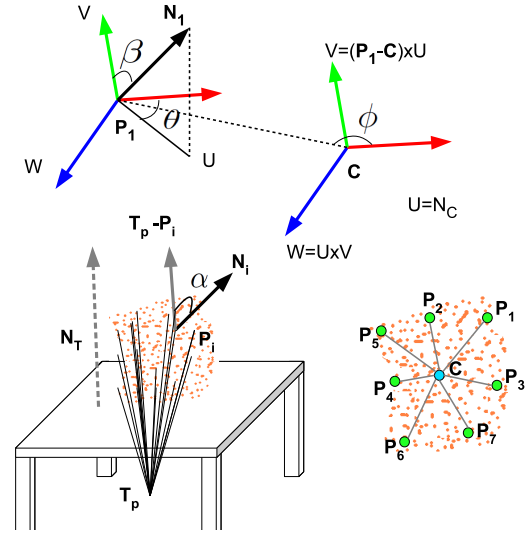


Fig. 4. Graphical representation of the two components of TPFH. TOP: eFPFH component with β, θ , and ϕ shown. V, W , and U represent the coordinates of the point C (centroid), with respect to the normal of the C and the position of C and P_1 . N_1 is the normal of point P_1 . BOTTOM: Table component with α shown. T_P is the normal 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 .



Fig. 5. LEFT: Article segmented from background. MIDDLE: The binary boundary of the article. RIGHT: 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.

invariant. Figure 6 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 [26] to calculate 100 codewords. These codewords provide each article with a 100 element feature vector that represents the local texture information.

2) *FPFH*: The FPFH, Fast Point Feature Histogram [24], 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 7 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 equation (10). FPFH is a histogram of 33 values, i.e. three

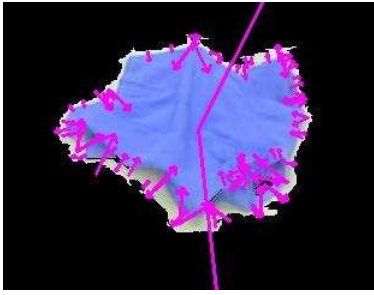


Fig. 6. Resulting SIFT features overlaid on the image. The arrows represent the orientation and scale (magnitude) of each SIFT descriptor.

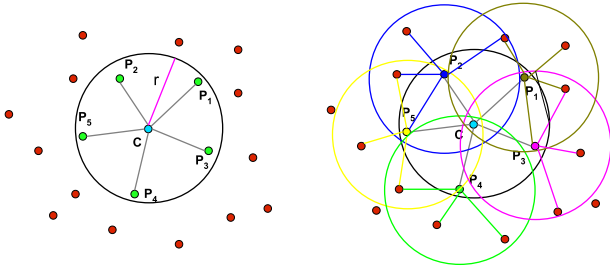


Fig. 7. LEFT: Example of the SPFH of a single point. An area of radius r encircles 5 neighboring points for this example. RIGHT: An example of the FPFH of the current point along with surrounding SPFHs used in the reweighting scheme. Each neighboring SPFH is color coded to represent the surrounding connections within the various regions.

sets of 11 bins for the three different orthogonal planes of XY, YZ , and XZ .

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

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. Concatenated, this yields 200 values for the local features. Then concatenated with global features yields 534 values, which are then fed to the multiple one-versus-all SVMs. Each of the local and global level features are scaled from $0 \rightarrow 100$ before concatenation to allow for equal weighting among each type of feature.

F. Mid-level layers

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 17 values that correspond to each characteristic used to learn what attributes are needed to separate the differences between shirts and dresses. The 17 characteristics that were chosen are shown in Table II, along with the number of occurrences

| | | |
|--------------------|-------------------|-----------------------|
| Collar (16) | Graphic texts (7) | Graphic pictures (44) |
| Top buttons (8) | Dark Colored (40) | Colored (52) |
| White Colored (25) | Striped (10) | Patterns (7) |
| Ankle hem (5) | Shin hem (15) | Wrist hem (21) |
| Shoulder hem (15) | Bicep hem (59) | Round neck (73) |
| V-neck (7) | Elastic Band (5) | |

TABLE II

LIST OF 17 UNIQUE CHARACTERISTICS AND THE NUMBER OF OCCURRENCES FOR EACH CHARACTERISTIC THAT WERE CHOSEN TO DIFFERENTIATE CATEGORIES OF CLOTHING. THE NUMBER OF ARTICLES CONTAINING EACH CHARACTERISTIC IS INDICATED IN PARANTHESES.

of each characteristic in the database. The training procedure used for classifying characteristics is the 5-fold cross validation (CV) [5], which is described in section III.

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 that particular characteristic, 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 II-G. If the current percentage was higher or equal to the previously calculated percentage, then we zeroed that particular characteristic for the next iteration.

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**.

G. 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 three unique mean vectors, one for each category. Then the selection mask is multiplied by the characteristic vector to zero out any characteristics with a negative importance.

III. EXPERIMENTAL RESULTS

The proposed approach was applied to a laundry scenario to test its ability to perform practical classification. In each

Input: All characteristic vectors for each category

Output: Selection masks for each category

$SelectionMask = 2^{16} - 1;$

$RemoveMask = 2^{17} - 1;$

foreach Article i **do**

$New_Character =$

$(Current_Character \times SelectionMask) \cap$

$RemoveMask;$

if $Result(New_Character) \geq$

$Result(Current_Character)$ **then**

$RemoveMask = RemoveMask \cap$

$2^{17} - 2^i - 1;$

end

$SelectionMask = SelectionMask \gg 1;$

if $i = I$ **then**

$SelectionMask = SelectionMask + 2^{16};$

end

end

Algorithm 1: The process of **C-S** to determine the final selection mask for each category.

experiment, the entire database consisted of 85 shirts, 10 dresses, and 22 socks. 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 mid-level characteristics (attributes) in grouping clothing categories that are difficult to classify. First, we consider the baseline approach that uses SVMs, with a radial basis function kernel, to directly classify each category without the use of mid-level characteristics. Then, we introduce characteristics as an alternate mapping from low level features to high level categories.

Each experiment was conducted using the 5-fold cross validation (CV) [5] to completely test each stage of the algorithm. We used 80% of each characteristic / 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 characteristic / category and 80% of the database that had a 0, or *FALSE*, for that characteristic / category to make the training set, the rest of the database went to the test set. The database was organized into five equally-sized groups that share the same amount of characteristics and same amount of categories. The resulting values from each SVM ranges from $-1 \rightarrow +1$, with a deciding threshold of 0. For these experiments, five instances of each article are used to describe the feature vector. We also tested the approach with four, three, two, and one instance for each article. We have found that using five instances provides a higher overall true positive rate.

A. Baseline approach without mid-level 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

| Local Features | | | |
|---------------------------|--------------|-------------|--------------|
| | Shirt | Dress | Socks |
| Shirt | 98.82 | 1.18 | 0.00 |
| Dress | 100.00 | 0.00 | 0.00 |
| Socks | 0.00 | 13.64 | 86.36 |
| Average TPR = 61.73% | | | |
| Global Features | | | |
| | Shirt | Dress | Socks |
| Shirt | 57.65 | 2.35 | 40.00 |
| Dress | 60.00 | 0.00 | 40.00 |
| Socks | 36.36 | 18.18 | 48.45 |
| Average TPR = 34.37% | | | |
| Local and Global Features | | | |
| | Shirt | Dress | Socks |
| Shirt | 40.00 | 0.00 | 60.00 |
| Dress | 40.00 | 0.00 | 60.00 |
| Socks | 18.18 | 27.27 | 54.55 |
| Average TPR = 31.52% | | | |

TABLE III

RESULTS OF BASELINE SYSTEM USING SVM TO CLASSIFY OBJECTS. TPR STANDS FOR TRUE POSITIVE RATE.

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 three categories to choose from, then three SVMs are trained in order to provide a probability value for each input vector from the test set. For this type of multi-class 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 [8], [19], [23], [1] 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. Results are shown in Table III.

B. 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. The goal of this experiment is to determine the increase of overall performance with the addition of characteristics. Results are shown in Table IV. The training set still consists of the ground truth characteristics to compute the mean vectors for each category. The intermediary results of classifying the characteristics using local and global features range from 16% \rightarrow 96%, with an average of 63%.

C. L-C-H vs. 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

| Local Features | | | |
|---------------------------|--------------|--------------|-------------|
| | Shirt | Dress | Socks |
| Shirt | 41.18 | 58.82 | 0.00 |
| Dress | 40.00 | 60.00 | 0.00 |
| Socks | 0.00 | 90.91 | 9.09 |
| Average TPR = 36.76% | | | |
| Global Features | | | |
| | Shirt | Dress | Socks |
| Shirt | 15.29 | 84.71 | 0.00 |
| Dress | 40.00 | 60.00 | 0.00 |
| Socks | 13.64 | 86.36 | 0.00 |
| Average TPR = 25.10% | | | |
| Local and Global Features | | | |
| | Shirt | Dress | Socks |
| Shirt | 21.18 | 78.82 | 0.00 |
| Dress | 20.00 | 80.00 | 0.00 |
| Socks | 36.36 | 63.64 | 0.00 |
| Average TPR = 33.73% | | | |

TABLE IV
RESULTS OF **L-C-H** PROPOSED SYSTEM TO CLASSIFY OBJECTS.

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

TABLE V
RESULTS OF **L-C-S-H** PROPOSED SYSTEM TO CLASSIFY OBJECTS.

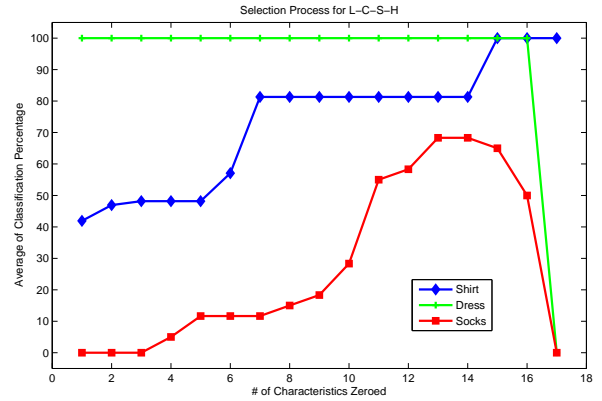


Fig. 8. Characteristic Selection Process for **L-C-S-H**: Results of removing a single redundant characteristic at each iteration. Starting from removing no characteristics and finishing with removing all characteristics.

categories. Results are shown in Table V. 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 8 illustrates the resulting TPR (True Positive Rate) as the most negatively important characteristic was 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 improve 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.

IV. CONCLUSION

We have proposed a novel approach to clothing classification (**L-S-C-H**) in which physical characteristics and selection masks are used to label and classify categories of clothing. This system uses a database of 2D color images and a 3D point cloud of clothing captured from a Kinect sensor. The paper is focused on determining how well this new approach improves classification over previous systems. The database that we use in our experiments contains articles of clothing whose configurations are difficult to distinguish their corresponding category.

The overall results of this new approach illustrates the critical importance of middle level information. The addition of middle level features, termed characteristics and selection masks, enabled the classification process to improve, on

average, by 27.65%. Using this approach could improve other types of classification / identification problems such as object recognition, object identification, person recognition, etc. 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 extensions to classify sub-categories, identities, gender, and the season of each article of clothing. The experiments identify a subset of characteristics that are useful for each category. Perhaps this approach could be used in pre-wash sorting. Three groups of darks, colors, and whites are separated before laundry is placed into the washing machine and then later dried. Future research could determine what other characteristics would be useful and how the classification would increase or decrease due to the characteristics used in this approach.

V. ACKNOWLEDGEMENTS

This research was supported by the U.S. National Science Foundation under grants IIS-1017007, and IIS-0904116.

REFERENCES

- [1] K. bo Duan and S. S. Keerthi, "Which is the best multiclass SVM method? an empirical study," in *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, 2005, pp. 278–285.
- [2] 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.
- [3] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," 2001.
- [4] 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.
- [5] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Prentice Hall, London, 1982.
- [6] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [7] P. Fitzpatrick, "First contact: An active vision approach to segmentation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2003, pp. 2161–2166.
- [8] 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.
- [9] M. Kaneko and M. Kakikura, "Planning strategy for unfolding task of clothes - isolation of clothes from washed mass -," in *13th Annual Conference of RSJ*, no. 1, 1996, pp. 455–456.
- [10] —, "Planning strategy for putting away laundry -isolating and unfolding task," in *Proc. of the 4th IEEE I.S. Assembly and Task Planning*, 2001, pp. 429–434.
- [11] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *Proceedings of the International Conference on Robotics and Automation*, May 2008, pp. 272–277.
- [12] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1377–1382.
- [13] 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.
- [14] Y. Kita, T. Ueshiba, E. Neo, and N. Kita, "Clothes state recognition using 3d observed data," in *Proceedings of the International Conference on Robotics and Automation*, May 2009, pp. 1220–1225.
- [15] —, "A method for handling a specific part of clothing by dual arms," in *Conf. on Intelligent Robots and Systems (IROS)*, 2009, pp. 3403–3408.
- [16] 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 IEEE IECON*, 2008, pp. 4180–4185.
- [17] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proceedings of the International Conference on Computer Vision*, 2009, pp. 365–372.
- [18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [19] J. Milgram, M. Cheriet, and R. Sabourin, "One Against One or One Against All: Which one is better for handwriting recognition with SVMs?" in *Tenth International Workshop on Frontiers in Handwriting Recognition*, Oct. 2006.
- [20] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Parametrized shape models for clothing," in *Proceedings of the International Conference on Robotics and Automation*, May 2011, pp. 4861–4868.
- [21] F. Osawa, H. Seki, and Y. Kamiya, "Unfolding of massive laundry and classification types by dual manipulator," *J of Advanced Computational Intelligence and Intelligent Informatics*, vol. 11, no. 5, pp. 457–463, 2007.
- [22] 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*, 2012, pp. 1703–1708.
- [23] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.
- [24] R. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," *Intelligent Autonomous Systems IAS*, 2009.
- [25] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2155–2162.
- [26] H. M. Wallach, "Topic modeling: Beyond bag-of-words," in *In Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 977–984.
- [27] B. Willimon, S. Birchfield, and I. Walker, "Rigid and non-rigid classification using interactive perception," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 1728–1733.
- [28] —, "Classification of clothing using interactive perception," in *International Conf. on Robotics and Automation (ICRA)*, 2011, pp. 1862–1868.
- [29] —, "Model for unfolding laundry using interactive perception," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [30] R. B. Willimon, "Interactive perception for cluttered environments," Master's thesis, Clemson University, Dec. 2009.