2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Real-time Pose Estimation of Deformable Objects Using a Volumetric Approach

Yinxiao Li†, Yan Wang†, Michael Case, Shih-Fu Chang, Peter K. Allen

Abstract— **Pose estimation of deformable objects is a fundamental and challenging problem in robotics. We present a novel solution to this problem by first reconstructing a 3D model of the object from a low-cost depth sensor such as Kinect, and then searching a database of simulated models in different poses to predict the pose. Given noisy depth images from 360-degree views of the target object acquired from the Kinect sensor, we reconstruct a smooth 3D model of the object using depth image segmentation and volumetric fusion. Then with an efficient feature extraction and matching scheme, we search the database, which contains a large number of deformable objects in different poses, to obtain the most similar model, whose pose is then adopted as the prediction. Extensive experiments demonstrate better accuracy and orders of magnitude speed-up compared to our previous work. An additional benefit of our method is that it produces a high-quality mesh model and camera pose, which is necessary for other tasks such as regrasping and object manipulation.**

I. Introduction

In robotics and computer vision, recognition and manipulation of deformable objects such as garments, are well-known challenging tasks. Recently, mature solutions to manipulating rigid objects have emerged and been applied in industry [3]. However, in the fabric and food industry, which involve a large number of deformable objects, there is still a large gap between the high demand for automatic operations, and the lack of reliable solutions. Compared with rigid objects, deformable objects are much harder to recognize and manipulate, especially because of the large variance of appearance in materials and the way they deform. This variance subsequently makes it difficult to establish a robust recognition pipeline to predict the *pose* of the deformable objects based on traditional visual sensors, such as regular cameras. However, newly emerged low-cost depth sensors such as Microsoft Kinect can provide accurate depth measurements. With this depth information, a robotic system is able to resolve the ambiguity of visual appearance better, and thus provide higher performance on recognition tasks.

Our interests are in detecting the pose of deformable objects such as garments as a part of a larger pipeline for manipulating these objects. Once the robot has identified the pose of the objects, it can then proceed to manipulate those objects, for tasks such as regarsping and garment folding.

† indicates equal contribution

Yinxiao Li, Michael Case, and Peter K. Allen are with the Department of Computer Science, Yan Wang is with the Department of Electrical Engineering, and Shih-Fu Chang is with the Department of Electrical Engineering and Department of Computer Science, Columbia University, New York, NY, USA. {yli@cs., msc2179, allen@cs.}columbia.edu {yanwang, sfchang}@ee.columbia.edu

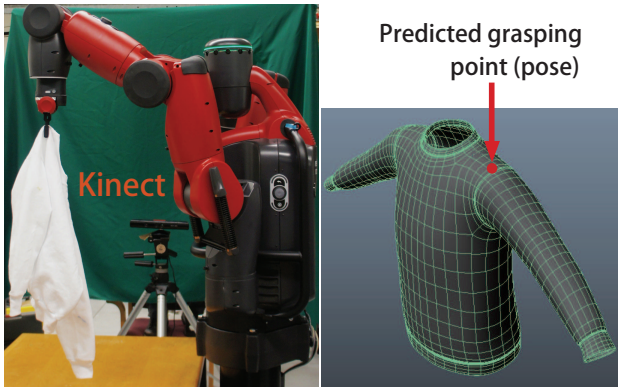

Fig. 1. Our application scenario: a Baxter robot grasps a sweater, and a Kinect captures depth images to recognize the pose of the sweater. The recognition result is shown on the right.

The method described in this paper is an improvement of our previous work [14] both in accuracy and speed. More specifically, our method can achieve real-time pose recognition with more accurate prediction of grasping point locations. Figure 1 shows a Baxter robot grasping a garment and predicting the grasping location (e.g. 2cm left of the collar). With this information, the robot is then able to proceed to following tasks such as regrasping and folding. The whole pipeline of garment folding is shown in the top row of Figure 2, whereas our work in this paper is focusing on pose estimation. The main idea of our method is to first accurately reconstruct a 3D mesh model from a low-cost depth sensor, and then compute the similarity between the reconstructed model and the models simulated offline to predict the pose of the object. Key contributions of our paper are:

- A real-time approach to reconstruct a smooth 3D model of a moving (e.g. rotating) deformable object from a noisy background without user interaction
- Formulation of the pose recognition as a real-time 3D shape matching task with a learned distance metric
- An automatic pipeline for building an offline database of deformable objects using a simulation engine that can be used for efficient data-driven pose recognition
- Experimental results with many different garments that show improved accuracy over our previous method [14], and orders of magnitude speed up yielding real-time performance for the pose estimation task

II. Related Work

While drawing more attention from the community, recognition and manipulation of deformable objects is relatively

978-1-4799-6934-0/14/$31.00 ©2014 IEEE

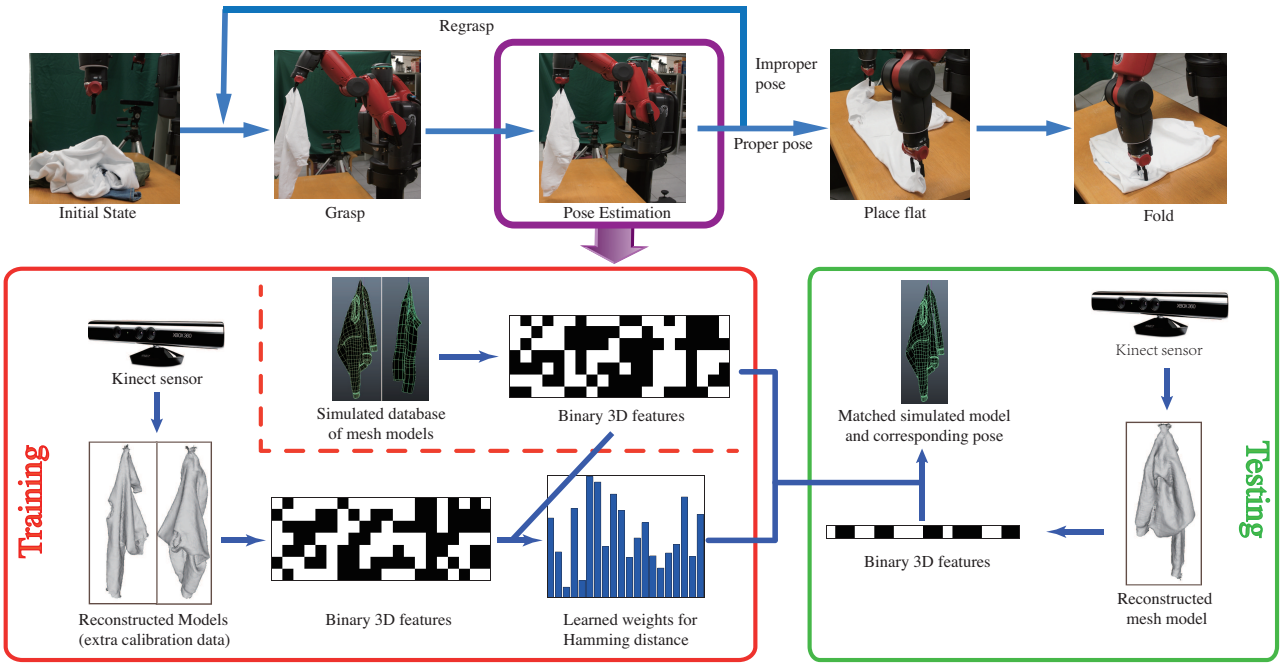

Fig. 2. Overview of our proposed pipeline for pose estimation of deformable objects. TOP ROW: The entire pipeline of dexterous manipulation of deformable objects. In this paper, we are focusing on the phase of pose estimation, as highlighted in the purple rectangle. If the recognition is not successful or the pose is improper for the following manipulation, the robot will regrasp the object and repeat the step of pose estimation. BOTTOM ROW: In the offline training stage (the red rectangle), we simulate mesh models of different types of garments in different poses, and learn a weighted Hamming distance from additional calibrated data collected from the Kinect. In the online testing stage (the green rectangle), we reconstruct a 3D model from the depth input, find the nearest neighbor from the simulated database with the learned distance metric, and then adopt the pose of the matched model as the output.

under-explored. Willimon *et al.* [24][25] proposed a method to classify the clothing using interactive perception. However, the heavy dependence on the color-based segmentation makes their method sensitive to the texture variance and limits the applicability. Wang *et al.* [22], Miller *et al.* [17], Maitin-Shepard *et al.* [8], and Cusumano-Towner *et al.* [5] have done work in clothing recognition and manipulation with a PR2 robot on tasks such as clothes folding. Their methods mainly focus on aligning the observed shape to a known template, which may suffer from self-occlusion and textureless input. For example, the work relying on corner detection [5][8] may fail on soft garments whose deformations are usually complicated and may produce many misleading corner-like features, and thus hard to reproduce the detected corners. Pose estimation of deformable objects is also a developing area. Kita *et al.* [10][11] have done a series of work on garment pose recognition. Their work is targeted at identifying the pose by registering the input to a shape template. However, according to the demonstrated experimental results, their methods work best when the extent of deformation is limited, while lacking further exploration on the practical cases where deformation is reasonably complex.

While most of the methods above are based on optical sensors including single or stereo cameras. In contrast, our most recent work [14] on recognition and pose estimation of deformable clothes uses a Kinect depth sensor. Our method does recognition on individual depth images and uses majority voting to get a comprehensive result. When the deformation is complicated, the inherent noise in the Kinect's sensor and the missing data due to the obstruction of self-occlusions may confuse the algorithm. However, running the whole pipeline on hundreds of input images slows down the method and thus limits the applicability. In this paper, rather than searching through a large number of discrete depth images, we instead reconstruct a 3D model of the garment in real time and use that to search a pre-computed database of simulated garment models in different poses.

Reconstructing a smooth 3D model from low-cost depth sensors is also closely related to our method. With the increasing popularity of Kinect sensor, there are various methods emerging in computer graphics such as KinectFusion and its variants [4][13]. Although these methods have shown success in reconstructing static scenes, they do not fit our scenario where a robotic arm is rotating the target garment about a grasping point. Therefore we first do a 3D segmentation to get the mask of the garment, and then invoke KinectFusion to do the reconstruction.

Shape matching is another related and long-standing topic in robotics and computer vision. On the 2D side, various local features have been developed for image matching and recognition [7][12][16], which have shown good performance on textured images. Another direction is shape-context based recognition [2][20], which is better for handwriting and character matching. On the 3D side, Wu *et al.* [26] and Wang *et al.* [21] have proposed methods to match patches based on 3D local features. They extract Viewpoint-Invariant Patches or the distribution of geometry primitives as features, based on which matching is performed. Thayanan-

than *et al.* [19], and Frome *et al.* [6] apply 3D shape-context as a metric to compute similarities of 3D layout for recognition. However, most of the methods are designed for noise-free human-designed models, without the capability to match between the relatively noisy and incomplete mesh model produced by Kinect and the human-designed models. Our method is inspired from 3D shape context [6], but provides the capability of cross-domain matching with a learned distance metric, and also utilizes a volumetric data representation to efficiently extract the features.

III. Method

Our method consists of two stages, the offline model simulation and the online recognition. In the offline model simulation stage, we use a physics engine [1] to simulate the stationary state of the mesh models of different types of garments in different poses. In the online recognition stage, we use a Kinect sensor to capture depth images of different views of the garment while it is being rotated by a robotic arm. We then reconstruct a 3D model from the depth input, extract compact 3D features from it, and finally match against the offline database for pose recognition. Figure 2 shows the framework of our method.

A. Model Simulation

Pose estimation of a deformable object such as a garment is challenging because of the large number of possible deformations. However, when grasped by a robotic arm, an object usually has limited deformation patterns, and the complexity is alleviated to within the capability of searching against an offline simulated database. To simulate such a database, we first collect a set of commercial 3D model of different garments. For each model, we select a set of points from the vertices of the model, and then use a physics engine to compute the mesh model of the garment in the stationary state as if a robotic arm were grasping at each selected point.

To select these grasping points, we first "unwrap" the mesh of the garment to a planar UV map, and then perform uniform sampling on it, as Figure 3 shows. The intuition behind is to obtain a piecewise linear mapping (rotating and minimal stretching in our case) on the vertices such that the result planar faces can preserve the size of the garment, with the final goal as to make uniform sampling on the 2D map result in uniformly distributed points in 3D. We use the Global/Local Mapping proposed in [15].

After the UV mapping step, we do uniform sampling (in terms of physical size) on the mapped plane. The grasping points are selected as the closest vertices to the sampled points, with one example shown in Figure 3. We employ similar physics simulation method described in our previous work [14], with the difference that the final outputs are mesh models instead of rendered depth maps. This simulation stage ends up with a set of mesh models with different garment types, material properties, and grasping points, which will be matched against a reconstructed model from a Kinect sensor.

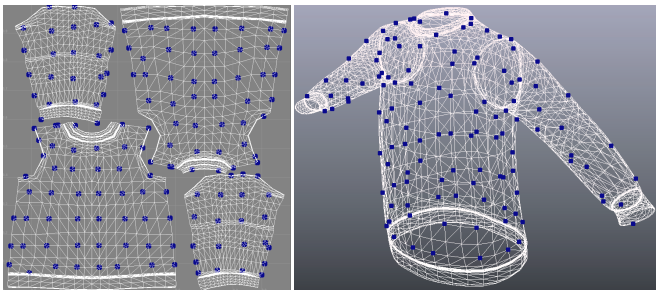

Fig. 3. An example of generating the set of grasping points for offline simulation. LEFT: The UV map of a sweater mesh model. Grasping points (the blue dots) are selected by finding the closest vertices to the uniformly sampled points. RIGHT: The original sweater model with the selected grasping points mapped back.

B. 3D Reconstruction

Given the model database described above, we now need to match the input against the database. As mentioned in Section II, direct recognition from depth images suffers from the problems of self-occlusion and sensor noise. This naturally leads to our new method of first building a smooth 3D model from the noisy input, and then performing matching in 3D. However, how to do such reconstruction is still an open problem. The existing approaches of obtaining high-quality models from noisy depth inputs usually requires the scene to be static. In our data collection settings, the target garment is being rotated by a robotic arm, which invalidates their assumptions. We propose to solve this problem by first segmenting out the garment from its background, and then invoke KinectFusion [18] to obtain a smooth 3D model. We assume that the rotation is slow and steady enough such that the garment will not deform further in the process.

Segmentation. Given the intrinsic matrix F_d of the depth camera and the ith depth image I_i, we are able to compute the 3D coordinates of all the pixels in the camera coordinate system with $\begin{bmatrix} x_{ci} & y_{ci} & z_{ci} \end{bmatrix}^T = F^{-1} d_i \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T$, in which (u_i, v_i) is the coordinate of a pixel in I_i, d_i is the corresponding depth, and (x_{ci}, y_{ci}, z_{ci}) is the corresponding 3D coordinate in the camera coordinate system.

Our segmentation is then performed in the 3D space. We ask the user to specify a 2D bounding box on the depth image $(x_{\min}, x_{\max}, y_{\min}, y_{\max})$ with a rough estimation of the depth of the garment $(z_{\min}, z_{\max})$. Given that the data collection environment is reasonably constrained, we find even one predefined bounding box works well. Then we adopt all the pixels having their 3D coordinates within the bounding box as the foreground, resulting in the masked depth images $\{I_i\}$ and their corresponding 3D points.

The 3D reconstruction is done by feeding the masked depth images $\{I_i\}$ into KinectFusion, while the unrelated surroundings are eliminated now, leaving the scene to reconstruct as static. This process can be done in real time. In addition to a smooth mesh, the KinectFusion library also generates a Signed Distance Function (SDF) mapping, which will be used for 3D feature extraction. The SDF is defined on any 3D point (x, y, z). It has the property that it is negative when the point is within the surface of the scanned object,

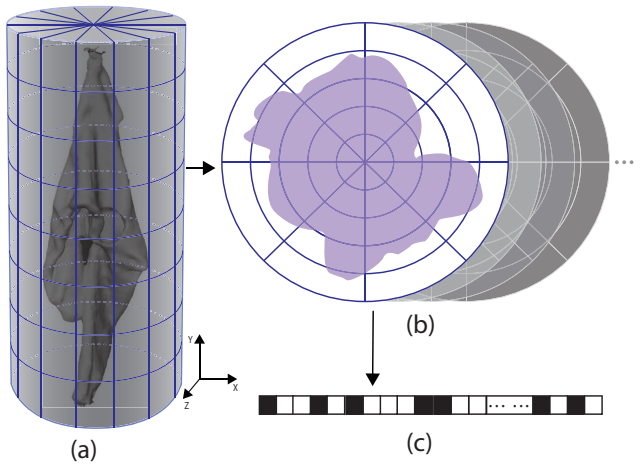

Fig. 4. Feature extraction from a reconstructed mesh model. (a) indicates that a bounding cylinder of a garment is cut into several layers. (b) shows a set of layers (sections). For each layer, we divide it into cells via rings and sectors. (c) shows a binary feature vector collected from each cell. Details are described in section III-C.

positive when the point is outside a surface, and zero when it is on the surface. We will use this function to efficiently compute our 3D features in the next subsection.

C. Feature Extraction

Inspired by 3D Shape Context, we design a binary feature to describe the 3D models. In our method, the features are defined on a cylindrical coordinate system fit to the hanging garment as opposed to traditional 3D Shape Context which uses a spherical coordinate system [6].

For each layer, as shown in Figure 4 top-right, we *uniformly* divide the world space into (R rings) $\times$ (Φ sectors) in a polar coordinate system, with the largest ring covering the largest radius among all the layers. The center of the polar coordinate system is determined as the mean of all the points in the highest layer, which usually contains the robot grasper. Unlike Shape Context, we do uniform instead of logarithm division of r, because Shape Context's assumption that cells farther from the center are less important no longer holds here. For each layer, instead of doing a point count as in the original Shape Context method, we check the Signed Distance Function (SDF) of the voxel which the center of the polar cell belongs to, and fill one (1) in the cell if the SDF is zero or negative (i.e. the cell is inside the voxel), otherwise zero (0). Finally, all the binary numbers in each cell are collected in order (e.g. with ϕ increasing and then r increasing), and concatenated as the final feature vector.

The insight is, to improve the robustness against local surface disturbance due to friction, we include the 3D voxels *inside* the surface in the features. Note we do not need to do the time-consuming ray tracing to determine whether each cell is inside the surface, but only need to look up their SDFs, thus dramatically speeding up the feature extraction.

Matching Scheme. Similar to Shape Context, when matching against two shapes, we conceptually rotate one of them and adopt the minimum distance as the matching cost, to provide rotation invariance. That is,

$$\text{Distance}(\mathbf{x}_1, \mathbf{x}_2) = \min_i \|R_i \mathbf{x}_1 \oplus \mathbf{x}_2\|_1, \tag{1}$$

in which $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{B}^{\Phi RN}$ are the features to be matched ($\mathbb{B}$ is the binary set $\{0, 1\}$), $\oplus$ is the binary XOR operation, and R_i is the transform matrix to rotate the feature of each layer by $2\pi/\Phi$. Recall that both features to be matched are compact binary codes. Such matching can be efficiently implemented by integer shifting and XOR operations, and is even faster than the Euclidean Distance given reasonable Φs (e.g. 10).

D. Domain Adaptation

Now we have a feature vector representation for each model in the simulated database and for the query. A natural idea is to find the Nearest Neighbor (NN) of the query in the database and transfer the metadata such as category and pose from the NN to the query. But NN on Euclidean distance does not work here because subtle differences in the deformation may cause dramatic Euclidean distance and essentially we are doing cross-domain retrieval. Given it is impractical to simulate every object with all possible materials, a common practice of cross-domain retrieval is to introduce a "calibration" step to adapt the knowledge from one domain (simulated models) to another (reconstructed models).

Weighted Hamming Distance. Similar with the distance calibration in [23], we use a *learned* distance metric to improve the NN accuracy, i.e.

$$\text{BestMatch}_{\mathbf{w}}(\mathbf{q}) = \arg\min_i \mathbf{w}^T \left(\hat{\mathbf{x}}_i \oplus \mathbf{q}\right), \tag{2}$$

in which $\mathbf{q}$ is the feature vector of the query, i is the index of models in the simulated database, and $\oplus$ is the binary XOR operation. $\hat{\mathbf{x}}_i = \hat{R}_i \mathbf{x}_i$ indicates the feature vector of the ith model, with $\hat{R}_i$ as the optimal R in Equation 1. The insight here is that we wish to grant our distance metric more robustness against material properties by assigning larger weights to the regions invariant to the material differences.

Distance Metric Learning. To robustly learn the weighted Hamming distance, we use an extra set of mesh models collected from Kinect as *calibration data*. The collection settings are the same as described in Section III-B and only a small amount of calibration data is needed for each category (e.g. 5 models in 28 poses for sweater model). To determine the weight vector $\mathbf{w}$, we then formulate the learning process as an optimization problem of minimizing the empirical error with a large-margin regularizer:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_j \xi_j \\ &\text{s.t. } \mathbf{w}^T\left(\hat{\mathbf{x}}_i \oplus \mathbf{q}_j\right) < \mathbf{w}^T\left(\hat{\mathbf{x}}_k \oplus \mathbf{q}_j\right) + \xi_j, \\ &\quad \forall j, \forall y_i = l_j, y_k \neq l_j, \\ &\quad \xi_j \geq 0, \end{aligned} \tag{3}$$

in which $\hat{\mathbf{x}}_i$ is the orientation-calibrated feature of the ith model (from the database), with y_i as the corresponding ground truth label (i.e. the index of the pose). $\mathbf{q}_j$ is the

extracted feature of the jth training model (from Kinect), with l_i as the ground truth label. We wish to minimize $\sum_i \xi_i$, which indicates how many wrong results the learned metric $\mathbf{w}$ gives, with a quadratic regularizer. C controls how much penalty is given to wrong predictions. This is a non-convex and even non-differentiable problem. Therefore we employ the RankSVM [9] to obtain an approximate solution.

Knowledge Transfer. Given the learned $\mathbf{w}$, in the testing stage, we then use Equation 2 to obtain the nearest neighbor of the query model. We directly adopt the grasping point of the nearest neighbor, which is known from the simulation process, as the final prediction.

IV. Experimental Results

We used a series of experiments to demonstrate the effectiveness of the proposed method and justify the components. We tested our method on a dataset of various kinds of garments collected from practical settings, and then quantitatively compared the result with our previous work for garment pose estimation described in [14]. To evaluate our results, the geodesic distance on the garment between the predicted grasping point and the ground truth is computed, together with the running time. Experimental results demonstrate that our method is able to achieve both higher accuracy and orders of magnitude speed-up.

A video of our experimental results is online at `http://www.cs.columbia.edu/~yli/3DVol.html`.

A. Dataset

We collect a dataset for general evaluation of pose recognition of deformable objects based on depth image as inputs. The dataset consists of three parts, a training set, a testing data set, and a calibration set. The training set is the simulated mesh models of different types of garments in different poses, as introduced in Section III-A. We bought 3 commercial-quality mesh models – sweaters, pants and shorts, and simulate each with $80-120$ grasping points (different types of garments have different number of grasping points depending on surface area and model complexity). Since all of our garment candidates are symmetric in front and back, left and right, we only adopt those grasping points on one fourth of surface over the whole garment to remove duplicates, ending up with 68 grasping points, each with a corresponding simulated mesh models.

To collect the testing set, we use a Baxter robot, which is equipped with two arms with seven degrees of freedom. A Kinect sensor is mounted on a horizontal platform at height of 1.2 meters to capture the depth images, as shown in Figure 1. With this setting, we collect data at the same grasping points of the training set, and then use our 3D reconstruction algorithm as introduced in Section III-B to obtain their mesh models. For each grasping point of each garment, the robot rotates the garment 360 degrees around 10 seconds while the Kinect captures at 30fps, which gives us around 300 depth images for each garment/pose. This results in a test set of 68 mesh models, with their raw depth images.

Given we also need to learn/calibrate a distance metric from extra data from Kinect, we collect an extra small amount of data with the same settings as the calibration data, only collecting 5 poses for each garment. A weight vector $\mathbf{w}$ is then learned from this calibration data for each type of garment as introduced in Section III-D.

B. Qualitative Evaluation

We demonstrate some of the recognition results in Figure 5 in the order of color image, depth image, reconstructed model, predicted model, ground truth model, and predicted grasping point (red) vs. ground truth grasping point (yellow) on the garment. From the figure, we can first see that our 3D reconstruction is able to provide us with good-quality models for a fixed camera capturing a dynamic scene. And our shape retrieval scheme with learned distance metrics is also able to provide reasonable matches for the grasping points. Note that our method is able to output a mesh model of the target garment, which is critical for the subsequent operations such as path planning and object manipulation.

C. Quantitative Evaluation

We first introduce some implementation details of our method, and then provide quantitative evaluations.

Implementation Details. In the 3D reconstruction, we set $X = 384$, $Y = Z = 768$ voxels and the resolution of the voxels as 384 voxels per meter to obtain a trade-off between resolution and robustness against sensor noise. In the feature extraction, our implementation adopts $R = 16, \Phi = 16, N = 16$ in the feature extraction as an empirically good configuration. That is, each mesh model gives a $16 \times 16 \times 16 = 4096$ dimensional binary feature. We set the penalty $C = 10$ in Equation 3.

Geodesic Error. For each input garment, we compute the geodesic distance of the predicted point and the ground truth, which we will refer as *Geodesic Error* in the following text, for evaluation. The distribution and the mean of the Geodesic Error are used as the evaluation protocol, and compared our method with our previous method [14]. Since our previous method uses depth images as input, for a fair comparison, we feed all the depth images to our previous algorithm for each pose of each garment.

A comparison of the distribution of the Geodesic Error is plotted in Figure 6. The total grasping points for sweater, jeans, and shorts are 28, 20, 20, respectively. We can clearly see that our method outperforms our previous method [14] in all the different garment types. Our method is benefited from the 3D reconstruction step, which reduces the sensor noise and integrates the information of each frame to a comprehensive model and thus leads to better decisions. Among three types of garments, recognition of shorts is not as accurate as the other two. One possible reason is that many of the shapes from different grasping points look similar. Even for human observers, it is hard to distinguish them.

To prove that the domain adaptation is a necessary step in our proposed method, we also test the Geodesic Error of our method without domain adaptation. The mean of the

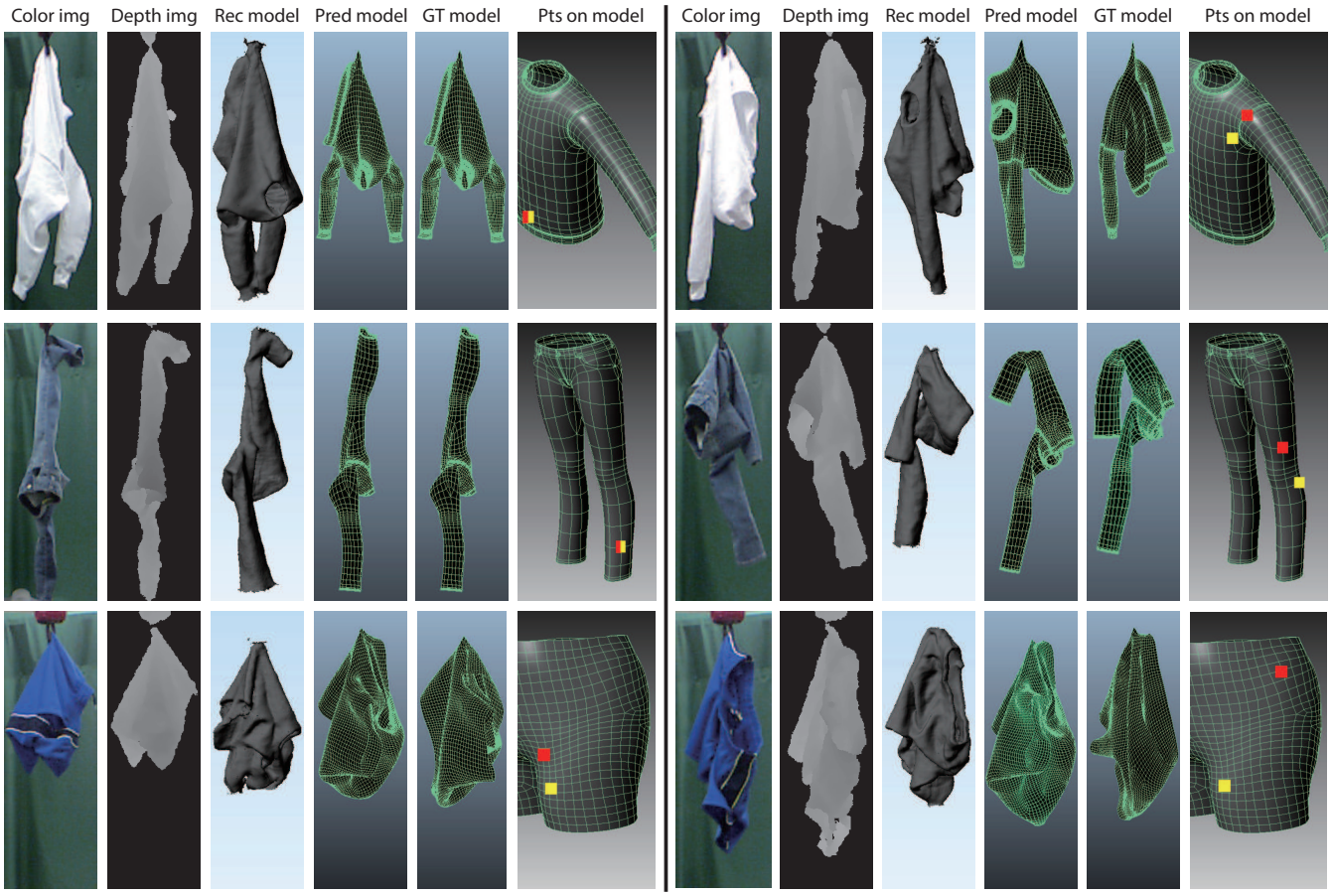

Fig. 5. Visual examples of the pose recognition result of our method. The garment is picked up via a griper of the Baxter robot. From left to right, each example shows the color image, input depth image, reconstructed model, matched simulated model, ground truth simulated model, and the predicted grasping points (red) marked on the model with the ground truth (yellow). The example shown in the bottom right shown here is considered as a failure example, which may be because of the uninformative deformation shape. Note our method does not use any color information. (Best viewed in color)

Geodesic errors of different method on different garments is illustrated in Table II. We can see that our method without the domain adaptation cannot beat our previous work, which verifies our motivation of introducing the cross-domain learning. When combined with the learned distance metric, our method is able to achieve lower Geodesic Error than results from [14].

Running Time. In addition, we also compare the processing time of our method compared to our previous method, which uses individual depth images. The time is measured on a PC with an Intel i7 3.0 GHz CPU, and shown in Table I. We can see that our method demonstrates orders of magnitude speed-up against the depth-image based method, which verifies our advantages from the efficient 3D reconstruction, feature extraction and matching. The main bottleneck of our previous method is SIFT extraction and sparse dictionary learning. Our method also shows better stability in running time, especially on the shorts input, while our previous method requires more time, especially when the depth input has rich textures.

TABLE I. Average running time in seconds of our method and our previous method, with the input of different garment types.

Garment	Previous Method	New Method
Sweater	46	**0.30**
Jeans	42	**0.20**
Shorts	71	**0.22**

TABLE II. Comparison on average Geodesic Error for different types of garments. The unit is cm. Ours (No DA) stands for our method without domain adaptation.

Garment	Previous Method	New Method (No DA)	New Method
Sweater	16.05	18.64	**13.61**
Jeans	10.89	14.31	**9.70**
Shorts	22.44	25.78	**17.82**

D. Application on Category Classification

A basic assumption of our method is that the category of the garment is known beforehand, so that we only need to search within the training data of the same category. But our method of Nearest Neighbor (NN) search can also be used to predict the category of the garment. Therefore we also test the performance of our method on this task, by searching the NN within the entire training set instead of only the part with the same category. By adopting the NN's category as the prediction, we are able to compute the classification

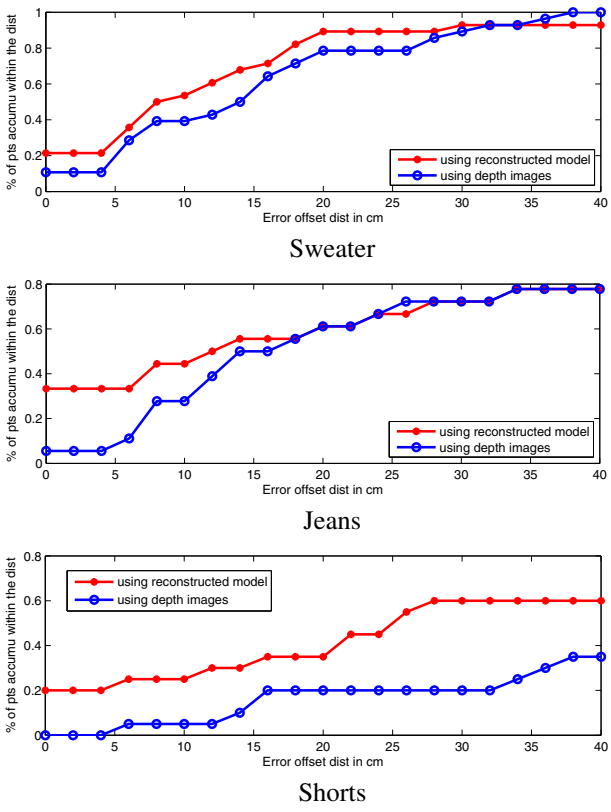

Fig. 6. Quantitative comparison of the proposed method (using reconstructed model) and our previous method (using individual depth images) [14]. The x axis is the Geodesic Error, and the y axis is the percentage of the input grasping points which give Geodesic Error smaller than the corresponding x. The results of a sweater, a pair of jeans, and shorts are shown from top to bottom, with maximum distance between any grasping points as 75cm, 65cm, and 55cm respectively.

TABLE III. Classification accuracy of our method on the task of garment categorization.

	Sweater	Jeans	Shorts
Accuracy	85.7%	70.0%	90.0%

accuracy for evaluation, as shown in Table III. We can see that our method is able to produce reasonable categorization results even without special optimization on the task.

V. Conclusion and Future Work

In this paper, we propose a novel volumetric approach for the problem of pose recognition of deformable objects using a low-cost depth sensor. We first reconstruct a mesh model from the depth images, extract a volumetric 3D feature from the model, and then match it against a simulated database with a learned distance metric to find the nearest neighbor, whose grasping point will be adopted as the prediction. Experiments demonstrate superior effectiveness and efficiency of our approach against our most recent work. These experiments assumed that each garment category was already known. We believe we can learn the category as well as the pose using an extension of this method.

Our work can be extended in several directions. Color and texture are possible to be added to the features to further improve the recognition accuracy. In addition, an accurate and fast method for pose estimation of deformable objects may benefit a variety of practical applications such as clothes folding, which will be much easier once the robot has an accurate mesh model and the grasping point. Our future focus will be on integrating richer information from the Kinect sensor to not only make our pose recognition more robust, but also improve tasks such as regrasping and folding.

Acknowledgments We'd like to thank Prof. E. Grinspun, J. Weisz, A. Garg, and Y. Yue for many insightful discussions. We'd also like to thank NVidia Corporation and Takktile LLC for the hardware support. This work is supported by NSF grant 1217904.

References

[1] Maya, http://www.autodesk.com/products/autodesk-maya/.
[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(24), April 2002.
[3] T. Brogrdh. Present and future robot control developmentan industrial perspective. *Annual Reviews in Control*, 31(1):69 – 79, 2007.
[4] J. Chen, D. Bautembach, and S. Izadi. Scalable real-time volumetric surface reconstruction. *SIGGRAPH*, 32(4):113:1–113:16, July 2013.
[5] M. Cusumano-Towner, A. Singh, S. Miller, J. F. OBrien, and P. Abbeel. Bringing clothing into desired configurations with limited perception. In *Proc. ICRA*, 2011.
[6] A. Frome, D. Huber, R. Kolluri, T. Blow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. ECCV*, pages 224–237, 2004.
[7] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *PAMI*, 1993.
[8] J. Lei J. Maitin-Shepard, M. Cusumano-Towner and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Proc. ICRA*, 2010.
[9] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. KDD*, pages 133–142, 2002.
[10] Y. Kita and N. Kita. A model-driven method of estimating the state of clothes for manipulating it. In *Proc. WACV*, 2002.
[11] Y. Kita, T. Ueshiba, E-S Neo, and N. Kita. Clothes state recognition using 3d observed data. In *Proc. ICRA*, 2011.
[12] L. Latecki, R. Lakamper, and T. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. CVPR*, 2000.
[13] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ToG (SIGGRAPH Asia)*, 32(6), November 2013.
[14] Y. Li, C-F Chen, and P. K. Allen. Recognition of deformable object category and pose. In *Proc. ICRA*, 2014.
[15] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler. A local/global approach to mesh parameterization. In *SGP*, 2008.
[16] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.
[17] S. Miller, J. Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel. A geometric approach to robotic laundry folding. *IJRR*, 2012.
[18] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011.
[19] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, 2003.
[20] Z. Tu and A. Yuille. Shape matching and recognition: Using generative models and informative features. In *Proc. ECCV*, 2004.
[21] J. Wang, L. Yin, X. Wei, and Y. Sun. 3d facial expression recognition based on primitive surface feature distribution. In *Proc. CVPR*, 2006.
[22] P-C Wang, S. Miller, M. Fritz, T. Darrell, and P. Abbbeel. Perception for the manipulation of socks. *Proc. IROS*, 2011.
[23] Y. Wang, R. Ji, and S.-F. Chang. Label propagation from imagenet to 3d point clouds. In *Proc. CVPR*, June 2013.
[24] B. Willimon, S. Birchfield, and I. Walker. Classification of clothing using interactive perception. In *Proc. ICRA*, 2011.
[25] B. Willimon, I. Walker, and S. Birchfield. A new approach to clothing classification using mid-level layers. In *Proc. ICRA*, 2013.
[26] C. Wu, B. Clipp, X. Li, J-M Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches. In *Proc. CVPR*, 2008.