A CRF that Combines Touch and Vision for Haptic Mapping

Ashwin A. Shenoi, Tapomayukh Bhattacharjee, and Charles C. Kemp

***Abstract*— Robots could benefit from maps that represent haptic properties of their surroundings. By touching locations with tactile sensors, robots can infer haptic properties of their surroundings, but touching all locations would be prohibitive. We present an algorithm that uses touch and vision to efficiently produce a dense haptic map. Our approach assumes that surfaces near a robot that are visually similar are more likely to have similar haptic properties. Given an image and sparse haptic labels, our algorithm uses a dense conditional random field (CRF) to produce a haptic map with labels for all image pixels. In an evaluation using images with idealized haptic labels, our algorithm substantially outperformed a previous algorithm. It also enabled a real robot to label leaves and trunks after reaching into artificial foliage. In addition, we show that our algorithm can use a convolutional neural network (CNN) for material recognition from Bell et al. [1] that we modified and fine-tuned. This CNN provides estimated probabilities for haptic labels using vision alone, which enables the algorithm to infer haptic labels before the robot makes contact with anything. In our evaluation, using this CNN further improved performance.**

I. Introduction

Robots could benefit from maps that represent how their local surroundings feel. For example, a robot might choose not to slide against a hard, rough surface to avoid damaging itself, or it might choose to compress a soft material to gain access to a location. In this paper, we present a method by which robots can use touch and vision to efficiently produce a haptic map. We define a haptic map as a set of pairs associating locations with haptic labels [2], and haptic labels as labels that represent properties of a location that can be inferred via tactile sensing.

Because of the inherently local nature of tactile sensing, a naive approach to haptic mapping would require that the robot make physical contact with each and every location of interest. This would be energetically expensive and time consuming. By using touch and vision, robots have the potential to haptically map their surroundings with greater efficiency. Our approach assumes that visible surfaces near a robot that are visually similar are more likely to have similar haptic properties. In our previous work [2], we introduced an iterative algorithm to infer dense haptic labels over a visible surface using sparse haptic labels. In this work, we introduce an improved algorithm (See Figure 2) to tackle the same problem using a dense conditional random field (CRF) [3]. We also show that our methods give better results using probabilities generated by a visual material recognition system before the robot makes contact. The visual material recognition system uses a convolutional neural network (CNN) developed by Bell et al. [1] that we fine-tuned.

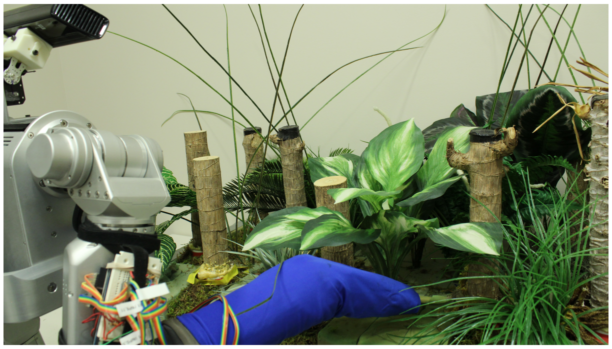

Fig. 1: *A robot DARCI, equipped with a tactile-sensing sleeve (blue) and a Kinect, reaching into the cluttered foliage environment*

We describe our algorithm in Section III. We analyzed the performance of our algorithm through simulated trials (see Section IV-A). We compared the performance of our algorithm using a uniform probability distribution versus probabilities obtained using the visual material recognition system (see Section IV-B) before the robot makes contact. We also evaluated our algorithm with a real robot in a cluttered foliage environment, as shown in Fig. 1 (see Sections V & VI).

II. Related Work

Researchers have worked on various ways of inferring properties of the environment using vision, tactile sensing, or a combination of both. Knowledge of the material properties of an object could help a robot deal with novel objects in the environment.

A. Material recognition using tactile sensing

Robots may make contact with various objects in the environment while performing manipulation tasks. Tactile sensing enables robots to gain information about object characteristics such as surface texture [6], [7], stiffness [8] and temperature [9]–[12]. These properties have been shown to be useful in material recognition [6], [7], [11], [12]. In addition, there have been extensive studies on material recognition using tactile sensing with features such as contact forces, contact motion, contact pressure, contact vibration etc. [13]–[16]. Tactile sensing only provides these labels to regions of contact. In this work, we use tactile information from these points of contact and couple it with information from vision to infer properties of the rest of the scene.

A. A. Shenoi, T. Bhattacharjee, and C. C. Kemp are with the Healthcare Robotics Lab, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology,

*A. A. Shenoi is the corresponding author {ashenoi@gatech.edu}.

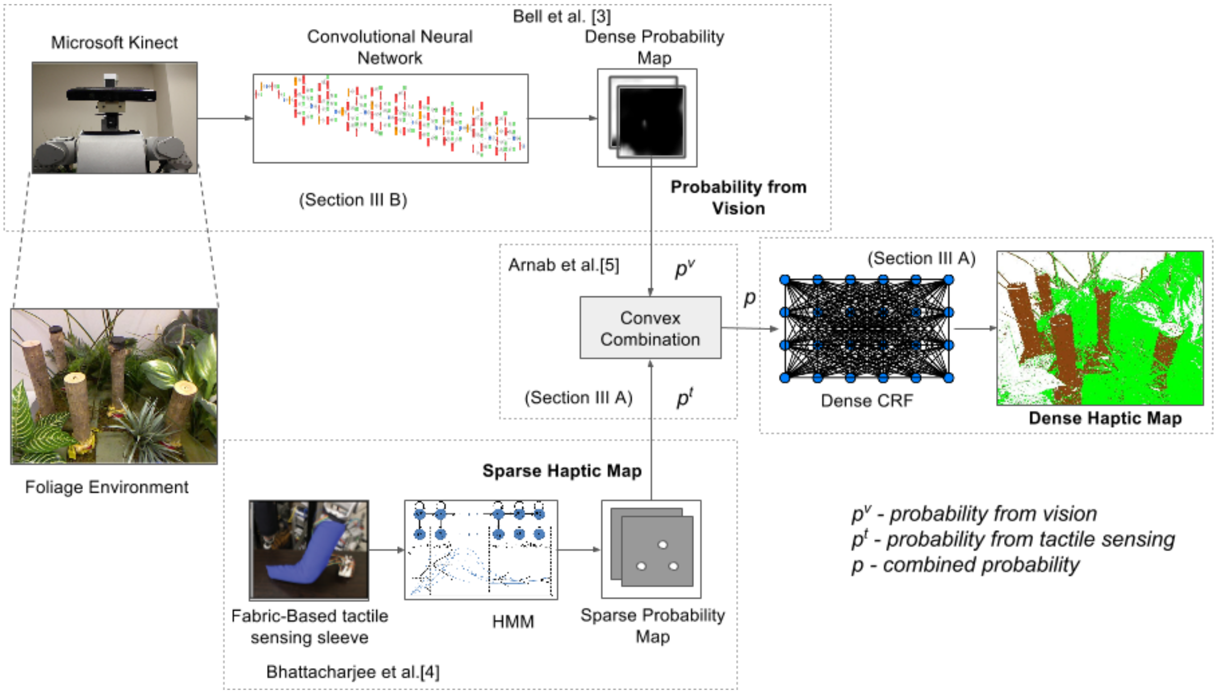

Fig. 2: *Integrating tactile sensing and vision for material recognition. The vision pipeline is processed by a fine-tuned convolutional neural network originally trained on the MINC dataset by Bell et al. [1]. We use a fabric based tactile sensing sleeve and HMMs [4] to classify points of contact. We combine the probabilities from the two modalities by taking a convex sum inspired by Arnab et al. [5]. We use a dense CRF to predict labels for each pixel from the combined probability map.*

B. Material recognition using vision

Vision has been used for texture recognition [17], [18]. Recent work has shown that vision can also be used for material recognition tasks [1], [5], [19]–[21]. Bell et al. [1] introduced a large scale database, Materials in Context Database (MINC), that has 23 material categories. They also introduced a framework that combines a convolutional neural network with a fully connected conditional random field to produce pixel level material labeling of the scene with 73.1% mean class accuracy. In this work we use the convolutional neural network model trained by Bell et al. [1] for the visual perception system.

CRFs are commonly used in vision problems to simultaneously segment and assign labels to each pixel in multi-class labeling problems [22]–[24]. Arnab et al. [5] used a joint dense CRF model to augment dense visual cues with sparse auditory cues to estimate dense object and material labels. While a basic CRF uses a pairwise potential term that incorporates local smoothing term, a dense CRF incorporates a pairwise potential between each individual pair of pixels, which enables long range interaction between pixels. This is useful for our task as it helps incorporate our notion that visually similar and spatially proximal points have similar labels and at the same time enables propagation of the information to spatially distant points.

C. Integration of touch and vision

Studies have shown that, under some conditions, humans can be modeled as combining visual and haptic information using a maximum-likelihood integrator [25]. The researchers proposed that humans integrate estimates of an environmental property through each individual sensory modality by performing a maximum likelihood estimator. Some early work in integrating vision and haptics [26]–[28] integrated information from the two modalities to build models of objects.

Allen [27] used vision to first determine objects of interest which the robot then explored using tactile sensing. The data from the two modalities were integrated to build a model that was compared with a model database to recognize the object. Stansfield [26] presented a robotic perceptual system that used vision to segment objects, and then haptically explored them to build a model of the object. Hosoda et al. [29] used a Hebbian network to learn consistency between data from a camera and tactile sensors to identify slip. Zytkow and Pachowicz [28] used vision and touch to learn object manipulation tasks. Luo et al. [30] combined vision and tactile sensing to localize the local point of contact by matching tactile feature with the visual map. In this work, we propose the use of dense CRFs to integrate the material classification predictions from tactile sensing and those made using vision to generate labels for the entire scene.

Kroemer et al. autonomously inferred low-dimensional representations from contact vibration tactile data by sliding a tactile sensor on 26 rich multi-scale surfaces of 17 different materials. They used both tactile and vision data in the training phase and created a mapping matrix, which they used in the testing phase with only tactile data [31]. Ueda et al. [32] used vision to observe the deformation of an object after interacting with it and used this information to extract rheological properties of the object. Charniya and Dudul [33], used a lightweight plunger and an optical mouse to take the surface image to classify the material. Zheng et al. [34] used deep learning for surface material classification using surface texture images and time-series of acceleration data measured from scratching the surface. They used multiple convolutional neural networks, one with images as inputs and the other with spectrograms of acceleration signals as inputs and used a fully-connected layer to combine information from both.

Gao et al. [35] trained two CNNs for haptic and vision data and combined their output using a fusion layer to assign up to 24 haptic adjectives to an object. To use the highest-performing version of their algorithm, a robot would obtain images of the object from multiple views, and record tactile signals while touching the object with four exploratory behaviors (*hold*, *squeeze*, *slow slide*, and *fast slide*). This work is strongly related to ours. For example, their haptic adjectives could be considered a type of haptic label, and they make use of the same material recognition CNN from Bell et al. [1] that we do. However, they focus on assigning multiple haptic labels to a single isolated object that the robot has haptically explored. In contrast, we focus on assigning haptic labels to locations all around the robot to produce a haptic map. Our current algorithm assigns a single haptic label to each location with the notion that the haptic label could be inferred by touching the location.

D. Haptic Mapping of the scene

Haptic maps generated via active exploration [36]–[38] and incidental contact [39] tend to be sparse due to the local nature of tactile sensing. Our previous work [39] used the sparse haptic map generated by tactile sensing and demonstrated its usefulness in manipulation tasks. Our previous work [2] then generated dense haptic maps of the visible scene by combining sparse data from tactile sensing with vision data. We achieved this by introducing an iterative algorithm that incorporates the notion that visually similar objects may have similar haptic properties. We used this to infer dense haptic labels of the scene from points of contact, and used a joint space planner to reach multiple goal locations in a cluttered environment with just one plan. However, this algorithm did not leverage the potential for vision to predict the haptic labels before the robot makes contact using visual cues alone. In this paper, we introduce an improved algorithm to achieve the same goals.

III. Algorithm

Our proposed algorithm (Figure 2), uses a dense CRF (Section III-A) to generate dense haptic labels from probabilities estimated using tactile sensing and vision (Section III-B). In Section III-C, we compare the algorithm with our previous algorithm [2] and we describe the implementation details in Section III-D.

A. Generating dense haptic map

We use a dense conditional random field [3] in a manner similar to Arnab et al. [5] to obtain a dense haptic map. Given a dense probability map from the visual modality (p^v) and a sparse probability map from tactile sensing (p^t), we combine the two probabilities using a convex combination. This is inspired by Arnab et al. [5], who generated material labels using two separate modalities, vision and audio. They combine these two terms by taking their convex combination. We use this approach since the haptic labels are sparse, similar to the labels acquired by audio sensing in [5]. We combine the two terms as shown in (1)

$$p_i(x_i) = \begin{cases} w_{tv} p_i^v(x_i) + (1 - w_{tv}) p_i^t(x_i) & if\ haptic\ label \\ & is\ available, \\ w_v p_i^v(x_i) + (1 - w_v) U & otherwise, \end{cases} \tag{1}$$

where x_i is the haptic label of pixel i, $p_i^v(x_i)$ is the probability of the label estimated by a classifier trained to predict labels using vision (Section III-B) and $p_i^t(x_i)$ is the probability of the label estimated by tactile sensing (true labels in Section IV and labels generated by hidden Markov models (HMMs) [4] in Section V). Note that only one haptic label from a mutually exclusive set can be assigned to each pixel i. U is a uniform distribution. w_{tv} and w_v are weight parameters (which can take a value between 0-1) that determine the importance of each individual prediction. Our algorithm then assigns haptic labels to individual pixels using a dense CRF [3]. It finds a set of labels that minimizes the Gibbs energy function defined as follows:

$$E(x|I) = \sum_i \psi_i(x_i) + w_p \sum_{i<j} \psi_{ij}(x_i, x_j) \tag{2}$$

where $\psi_i(x_i)$ is the unary term and $\psi_{ij}(x_i, x_j)$ is the pairwise term that connects every pixel pair in the image. We use the unary and pairwise terms from Bell et al. [1]. w_p is the weight for the pairwise term.

$$\psi_i(x_i) = -\log p_i(x_i) \tag{3}$$

$$\psi_{ij}(x_i, x_j) = \delta(x_i \neq x_j) k(f_i - f_j) \tag{4}$$

In (4), δ is a label compatibility term which introduces a penalty if two pixels are assigned different labels (See eq. (5). k is a Gaussian kernel.

$$\delta(x_i \neq x_j) = \begin{cases} 1 & if\ x_i \neq x_j, \\ 0 & otherwise, \end{cases} \tag{5}$$

The feature f_i used in [1] is the color (I^L, I^a, I^b) represented in $L^*a^*b^*$ color space and position (p^x, p^y) of each pixel:

$$f_i = [\frac{p_i^x}{\theta_p d}, \frac{p_i^y}{\theta_p d}, \frac{I_i^L}{\theta_L}, \frac{I_i^a}{\theta_{ab}}, \frac{I_i^b}{\theta_{ab}}] \tag{6}$$

where θ_{ab}, θ_L and θ_p are constants and d is the smaller input image dimension. To summarize,

1) We use the the dense CRF framework used by Bell et al. [1]
2) We use the formulation used by Arnab et al. [5] to combine the probabilities from the two modalities.

B. Probability from vision

To generate a probability distribution for material recognition based on the scene before contact, we use a convolutional neural network trained for material recognition in images by Bell et al. [1]. Bell et al. [1] trained the network on the MINC database released by the same authors. The images in the MINC dataset are annotated with 23 material categories. For this work, we fine-tuned this network to recognize 8 material categories (*Ceramic, Paper, Plastic, Metal, Fabric, Wood, Glass and Other*) using patches extracted from various RGB-D datasets [5], [40]–[44]. These labels could reasonably be classified using tactile sensing [7], [11] and they better match the labels in the MINC database [1].

We annotated 228 images from these publicly available datasets with the 8 material categories mentioned above. We ensured that none of these 228 images used for fine tuning were part of the image dataset used for evaluation (Section IV). We then adopted the same procedure used by Bell et al. [1] to extract patches from these 228 images. Specifically, we used Poisson disk sampling to sample pixels in the images and extracted square patches centered around these points. The dimension of each patch was 23.1% of the smaller image dimension. We then fine-tuned the MINC CNN model using Caffe [45]. We replaced the last fully connected layer (originally with 23 outputs) with a fully connected layer with 8 outputs. Since our dataset is small, we froze the weights of all the layers till the inception (3b) layer. For the rest of the convolution layers, we used one tenth of the learning rate of the last layer. We used the stochastic gradient descent (sgd) method for optimization. We used a base learning rate of 0.001. We would step it down by a factor of 10 every 20000 iterations. We set the momentum(γ), which helps accelerate the learning, to 0.9. We also tried to fine-tune the values of the earlier layers, but this did not help improve the results with our dataset. We did the training on an Amazon AWS g2.2xlarge instance.

C. Comparison with our previous algorithm [2]

In our previous work [2], we introduced an iterative algorithm to infer dense haptic labels from a sequence of sparse haptic labels. Our previous algorithm achieved this by incorporating the notion that visually similar points are likely to have similar haptic labels. We maintained a list of points of contact and their associated color and haptic label. The dense haptic labels were assigned by finding similar points using a distance metric in the color space. In our previous algorithm, vision was only used to determine similar points. Our present algorithm also uses vision to estimate probabilities of haptic labels.

D. Implementation:

We implemented our algorithm in Python using the scikit-image [46], NumPy [47] and OpenCV [48] libraries. We used the Python code provided by Bell et al. [1], which uses Caffe [45] for building CNN and the C++ implementation of dense CRF released by Krähenbühl et al. [3].

TABLE I: Values of parameters used for simulations and experiments

Parameter	Simulations	Experiments
w_v	0.001	0.0
w_{tv}	0.001	0.01
w_p	80	50
θ_p	0.5	0.5
θ_L	3.0	3.0
θ_{ab}	0.5	0.5

TABLE II: Comparison of performance of our current algorithm with our previous algorithm [2].

Contact Points/ No. of Objects	*Pixels correctly labeled*	
	Current (*Avg.*±*StdDev*)%	*Previous* (*Avg.*±*StdDev*)%
5	81.14 ±15.02 %	63.08 ±19.71 %
10	85.12 ±11.68 %	69.48 ±18.26 %
15	87.58 ±9.73 %	71.98 ±17.28 %
20	89.20 ±8.68 %	73.84 ±17.02 %
25	90.72 ±7.62 %	75.11 ±16.18 %
30	91.59 ±6.89 %	75.67 ±16.09 %
35	92.37 ±6.16 %	74.98 ±16.91 %
40	**93.05 ±5.58 %**	76.02 ±16.26 %

TABLE III: Performance on different environments after 40 contact points per object.

Env. Type	F_1*score* $[0,1]$ (*Avg.*±*Std.Dev.*)	*Pixels correctly labeled* (*Avg.*±*Std.Dev.*)%
Low Clutter	**0.86 ±0.12**	**94.36 ±5.16 %**
High Clutter	0.72 ±0.13	90.28 ±5.42 %
Bed	**0.92 ±0.06**	**97.59 ±2.02 %**
Floor	0.92 ±0.10	96.85 ±3.77 %
Shelf	0.82 ±0.11	92.94 ±5.28 %
Sink Area	0.76 ±0.14	90.60 ±5.87 %
Table Top	0.82 ±0.14	93.10 ±5.45 %
Misc.	0.84 ±0.13	95.23 ±4.36 %

IV. Evaluation with Simulated Trials

A. Comparison with our previous algorithm

First, we compared the performance of our algorithm to our previous algorithm from [2] using the same evaluation procedure. We used the same set of 186 RGB-D images of indoor cluttered scenes suitable for robot manipulation tasks from various publicly available RGB-D datasets and the same set of haptic labels as in [2]. Note, for simulations, we assumed there is no uncertainty in the haptic label. For each image, we generated a pool of labeled pixels by randomly selecting 1000 labeled pixels from each segmented object in the image. We then randomly sampled 40 * N_i pixels without replacement from this pool, where N_i is the number of objects and i is the image. N_i had values that ranged from 1 object to 24 objects. We repeated this process for each of the 186 images, resulting in $\sum_{i=1}^{186} 40 * N_i = 52160$ labeled pixels in total. For each of the sampled points, we assumed that a patch of size 10x10 centered around this point had the same haptic label as the center pixel and updated the probability map for this patch. In our previous algorithm, we did not make this assumption and considered the color of the single pixel in the middle. Table I shows the values of the different parameters used in our simulations.

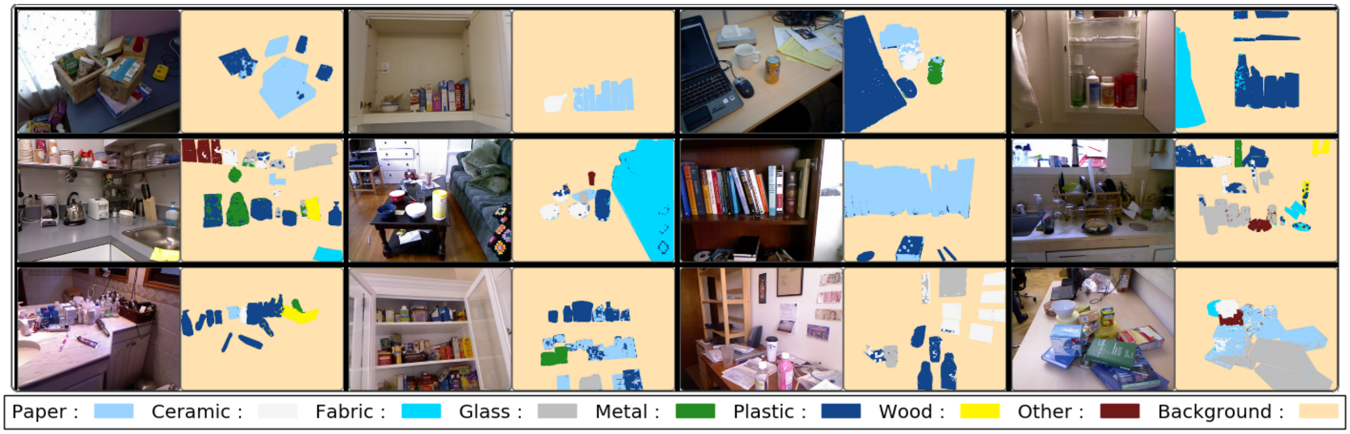

Fig. 3: *Simulation results of haptic categorization after 40 contact points per object for some example images from different publicly available datasets [41], [42], [49]–[51]. We chose indoor cluttered scenes suitable for robot-manipulation tasks. These examples show scenes from different environments with varying density of clutter.*

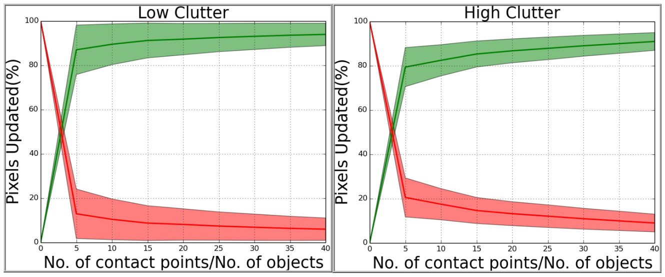

Fig. 4: *Percentage of pixels assigned correct/incorrect labels for different environment types. Green: Correct, Red: Incorrect*

To evaluate how our algorithm performed with more contacts with objects in the environment, we found the number of pixels that were correctly updated with each new point of contact. We ignored the background for our evaluation. Table II shows the results and compares it with the results from our previous algorithm. As the number of contacts increased, the rate at which the pixels were correctly updated decreased. Feedback-driven sampling, such as sampling from locations that have not yet been labeled, might result in improved performance. With a ratio of 40 contact points per object, the algorithm correctly updated an average of 93% of the object pixels in an image. Since there were 8602 pixels per object on average, 40 pixels per object is a relatively small portion of the visible scene. Note that with just 5 contact points per object, the algorithm correctly updated an average of 81% of pixels, which is higher than the results achieved for 40 contact points per object with our previous algorithm in [2]. Figure 3 shows various images from the dataset used in the simulation and the corresponding outputs.

1) Effect of Clutter: We classified the images in our dataset into two categories, low clutter and high clutter. We computed the $F_1 score$ and percentage of pixels updated with a ratio of 40 contact points per object for all images in each category. Table III and Fig. 4 show the results. Our algorithm performed better with low-clutter environments ($F_1 score = 0.86$) when compared to high-clutter environments ($F_1 score = 0.72$).

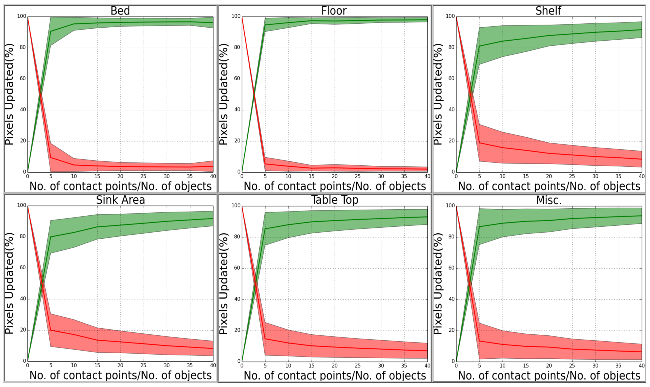

Fig. 5: *Percentage of pixels assigned correct/incorrect labels for different environment types. Green: Correct, Red: Incorrect*

2) Effect of Type of Environment: We also classified the images into 6 different scene-based categories. We computed the same performance measurements as in Section IV-A.1. Table III and Fig. 5 show the results. Note that the bed and floor environments were less cluttered than shelf and sink environments.

B. Effect of probability from vision–p^v

For this set of simulations, we made two changes to the procedure adopted in Section IV-A. First we redefined the haptic labels of the 186 images as *Ceramic, Paper, Plastic, Metal, Fabric, Wood, Glass and Other*. These labels could reasonably be classified using tactile sensing and they better match the labels in the MINC database [1]. Second, we randomly sampled the points of contact directly from the image instead of sampling from the pool as done in Section IV-A.

We evaluated the effect of probability from vision (p_v) by comparing the performance of a uniform probability distribution with the probability distribution generated using

TABLE IV: Effect of probability from vision on performance.

Contact Points	*Pixels correctly labeled*	
	Uniform (*Avg.*±*StdDev*)%	*From CNN* (*Avg.*±*StdDev*)%
0	15.27 ±25.66 %	**36.96 ±28.28 %**
100	84.94 ±16.08 %	88.68 ±9.21 %
200	87.12 ±15.47 %	91.41 ±7.17 %
300	88.6 ±15.11 %	93.07 ±5.9 %
400	89.63 ±14.93 %	94.26 ±4.99 %
500	90.43 ±14.83 %	95.16 ±4.32 %
600	91.05 ±14.79 %	95.89 ±3.74 %
700	91.59 ±14.76 %	96.47 ±3.29 %
800	92.0 ±14.75 %	96.92 ±2.9 %
900	92.32 ±14.74 %	97.32 ±2.55 %
1000	92.6 ±14.73 %	**97.63 ±2.3 %**

the convolutional neural network released by Bell et al. [1] that we modified and fine-tuned. We repeated the same simulation as described in IV-A using the same set of 186 images. We performed the simulation by initializing p^v to two different distributions described below:

1) Uniform Distribution: We assume that the robot has no knowledge of the class of the pixels. We assign equal probability ($p^v = 1/8$) to all classes.

2) Probability Distribution determined by CNN: We set p^v to the probability map of the image using the fine-tuned CNN described in Section III-B.

Table IV shows the results for the two different distributions. Before contact, the dense CRF using a uniform probability distribution assigns the correct labels to 15.27% of the pixels. The dense CRF using the probability distribution from CNN assigns the correct labels to 36.96% of pixels. We see improvement in the performance of the algorithm when it uses the probability distribution generated from the CNN.

V. Evaluation with a Real Robot

A. Experimental Setup

We used the humanoid robot DARCI (Fig. 1), a Meka M1 Mobile Manipulator, which includes a mobile base, a torso on a vertical linear actuator, and two 7-DoF arms. The mobile base and torso height remained fixed throughout our experiments. The right arm had a fabric based tactile-sensing sleeve [52]. The tactile-sensing sleeve has 25 discrete taxels. It records the contact force and we trained HMMs for haptic categorization [53]. The joints of the robot arm use series elastic actuators (SEAs) and have a real-time impedance controller with gravity compensation. This simulates low-stiffness visco-elastic springs at the robot's joints. We mounted a Microsoft Kinect on top of the torso. Our algorithm processes the data from the two sensors (Kinect and sleeve) as described in Section III. For our experiments, we used a system that runs Ubuntu 12.04 32-bit OS with a 3.5.0-54-generic linux kernel. It has 16 GB RAM and an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz × 8 processor. We used ROS Fuerte [54] for communicating with the RTPC on the robot DARCI. We used cv bridge [55] to convert between ROS images and OpenCV images. We used the GHMM toolkit [56] to implement and train the HMMs. We tested the algorithm in an artificial foliage environment. The environment is composed of trunks and leaves as seen in Figure 1. We used this setup in our previous work [2].

TABLE V: Performance of the algorithm on the foliage environment.

Number of reaches (*N*)	Average number of contact points	*Pixels correctly labeled* (*Avg.*±*StdDev*)%
0	0	27.76 ±1.57 %
1	6.67	58.36 ±20.35 %
2	7	65.37 ±27.07 %
3	7.33	65.37 ±27.07 %
4	7.33	65.37 ±27.07 %
5	7.33	65.37 ±27.07 %
6	12	86.87 ±10.97 %
7	16	87.04 ±10.73 %
8	**16.67**	**87.07 ±10.75 %**
9	17	84.46 ±14.44 %
10	22.67	82.52 ±9.70 %

B. Experimental Procedure

We programmed the robot DARCI to make 10 reaches into the foliage environment for each trial. We conducted 3 such trials with different configurations of the foliage (See Fig. 6). In total, the arm reached into 30 end-effector goal locations (5 pre-selected goal locations × 2 times × 3 trials). For each of the trials, we randomized the order in which the goal locations were selected. After reaching each goal location, the robot-arm came back to the initial starting location and then moved to the next randomly selected goal location. The initial starting location of the robot-arm was the same for all trials.

During each reach, the robot used our previously developed dynamic MPC controller [57] to reach the goal location with low contact forces. The robot made incidental contact with various points in the environment. We used forward kinematics to locate the contact points and transformed the coordinates of the points of contact to the image pixel coordinates using the camera properties and depth information. We ignored contacts beyond visible surface. We identified those contacts as contacts for which the estimated depth of the contact point was greater than the depth of the visible surface. We used trained left-right HMMs with 10 states and a uniform prior for our experiments. We had one HMM model each for trunk and leaf and classified the contact points as trunks or leaves based on maximum likelihood estimates. We used this information and the Kinect image to infer haptic properties for the rest of the scene using our algorithm.

VI. Experimental Results

We annotated the RGB image of the final scene after the robot completed 10 reaches into the environment. We only annotated the regions belonging to trunk or leaf and treated the rest as background. We used this annotated image as ground truth for our evaluation. Figure 6 shows the haptic maps generated after various trials. Ignoring the background, we evaluated the percentage of the pixels that our algorithm correctly assigns to trunks and leaves after each reach. Table I shows the values of the different parameters used for our experiments. Table V reports the results. The algorithm

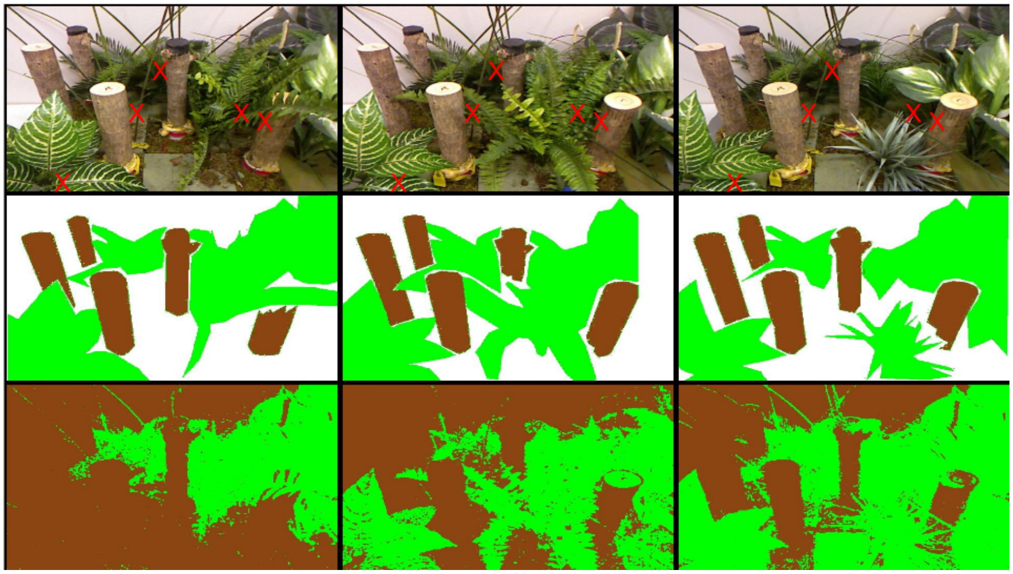

Fig. 6: *The top row shows the scene after the robot made 10 reaches. The middle row shows the annotated images which are the ground truth. The bottom row shows the corresponding haptic map. The trunks are marked with brown, the leaves are marked with green and the background is marked in white. We ignored the background for our evaluation. Red X's in the top row show the goal locations.*

assigned the correct labels to 82.52% of pixels that belong to trunks or leaves after 10 reaches. Note that, unlike in simulations, there is some uncertainty involved in the haptic labels generated by the tactile perception system during evaluations with the real robot. See the accompanying video for its real-time performance.

VII. Conclusion

We presented a dense CRF-based algorithm to obtain dense haptic maps across visible surfaces given sparse haptic labels. We based our approach on the notion that surfaces near the robot that look visually similar are more likely to feel similar to one another when touched. To evaluate our algorithm, we used idealized haptic labels with a collection of 186 indoor images pertinent to robot manipulation selected from various publicly available RGB-D datasets [2]. Our algorithm performed substantially better than our previous algorithm [2]. In general, it performed better for low-clutter scenes than for high-clutter scenes. As expected, with more haptic labels, our algorithm performed better at inferring the correct haptic labels across the scene. In addition, we found that using a probability distribution obtained from a CNN with vision data improved performance. Our algorithm obtained an average accuracy of 98% with the CNN versus 93% without, given 1000 contact points. Given 0 contact points, our algorithm achieved 37% with the CNN versus 15% without. We also evaluated our algorithm on a real robot reaching in artificial foliage. It assigned the correct label to 82.52% of pixels after 10 reaches.

Acknowledgment

We thank Zackory Erickson for his feedback. This work was supported in part by NSF Emerging Frontiers in Research and Innovation (EFRI) Award 1137229 and NSF Award IIS-1150157. We also thank Bell et al. [1], for providing their code and model to run their framework.

References

[1] S. Bell, P. Upchurch, N. Snavely, and K. Bala, “Material recognition in the wild with the materials in context database,” *arXiv preprint arXiv:1412.0623*, 2014.

[2] T. Bhattacharjee, A. A. Shenoi, D. Park, J. M. Rehg, and C. C. Kemp, “Combining tactile sensing and vision for rapid haptic mapping,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sep-Oct 2015.

[3] P. Krähenbühl and V. Koltun, “Parameter learning and convergent inference for dense random fields,” in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 513–521.

[4] T. Bhattacharjee, A. Kapusta, J. M. Rehg, and C. C. Kemp, “Rapid categorization of object properties from incidental contact with a tactile sensing robot arm,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, October 2013.

[5] A. Arnab, M. Sapienza, S. Golodetz, J. Valentin, O. Miksik, S. Izadi, and P. H. Torr, “Joint object-material category segmentation from audio-visual cues,” *British Machine Vision Conference (BMVC)*, 2015.

[6] N. Jamali and C. Sammut, “Material classification by tactile sensing using surface textures,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2336–2341.

[7] R. Li and E. H. Adelson, “Sensing and recognizing surface textures using a gelsight sensor,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1241–1247.

[8] Y. Murayama, C. E. Constantinou, and S. Omata, “Development of tactile mapping system for the stiffness characterization of tissue slice using novel tactile sensing technology,” *Sensors and Actuators A: Physical*, vol. 120, no. 2, pp. 543–549, 2005.

[9] G. J. Monkman and P. Taylor, “Thermal tactile sensing,” *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 3, pp. 313–318, 1993.

[10] C. H. Lin, T. W. Erickson, J. Fishel, N. Wettels, G. E. Loeb, *et al.*, “Signal processing and fabrication of a biomimetic tactile sensor array with thermal, force and microvibration modalities,” in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*. IEEE, 2009, pp. 129–134.

[11] T. Bhattacharjee, J. Wade, and C. C. Kemp, “Material recognition from heat transfer given varying initial conditions and short-duration contact,” *Proceedings of Robotics: Science and Systems, Rome, Italy*, 2015.

[12] T. Bhattacharjee, J. Wade, Y. Chitalia, and C. C. Kemp, "Data-driven thermal recognition of contact with people and objects," in *2016 IEEE Haptics Symposium (HAPTICS)*. IEEE, 2016, pp. 297–304.

[13] T. Bhattacharjee, J. M. Rehg, and C. C. Kemp, "Haptic classification and recognition of objects using a tactile sensing forearm," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 4090–4097.

[14] J. Hoelscher, J. Peters, and T. Hermans, "Evaluation of tactile feature extraction for interactive object recognition," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 310–317.

[15] T. Bhattacharjee, J. M. Rehg, and C. C. Kemp, "Inferring object properties from incidental contact with a tactile sensing forearm," *arXiv preprint arXiv:1409.4972*, 2014.

[16] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, T. Darrell, and K. J. Kuchenbecker, "Robotic learning of haptic adjectives through physical interaction," *Robotics and Autonomous Systems*, vol. 63, pp. 279–292, 2015.

[17] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International journal of computer vision*, vol. 43, no. 1, pp. 29–44, 2001.

[18] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 11, pp. 2032–2047, 2009.

[19] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a bayesian framework for material recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 239–246.

[20] D. Hu and L. Bo, "Toward robust material recognition for everyday objects." Citeseer.

[21] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3828–3836.

[22] C. Russell, P. Kohli, P. H. Torr, *et al.*, "Associative hierarchical crfs for object class image segmentation," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 739–746.

[23] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[24] B. Fulkerson, A. Vedaldi, S. Soatto, *et al.*, "Class segmentation and object localization with superpixel neighborhoods." in *ICCV*, vol. 9. Citeseer, 2009, pp. 670–677.

[25] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 6870, pp. 429–433, 2002.

[26] S. A. Stansfield, "A robotic perceptual system utilizing passive vision and active touch," *The International journal of robotics research*, vol. 7, no. 6, pp. 138–161, 1988.

[27] P. K. Allen, "Integrating vision and touch for object recognition tasks," *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 15–33, 1988.

[28] J. M. Zytkow and P. W. Pachowicz, "Fusion of vision and touch for spatio-temporal reasoning in learning manipulation tasks," in *1989 Advances in Intelligent Robotics Systems Conference*. International Society for Optics and Photonics, 1990, pp. 404–415.

[29] K. Hosoda, Y. Tada, and M. Asada, "Internal representation of slip for a soft finger with vision and tactile sensors," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1. IEEE, 2002, pp. 111–115.

[30] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Localizing the object contact through matching tactile features with visual map," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3903–3908.

[31] O. Kroemer, C. H. Lampert, and J. Peters, "Learning dynamic tactile sensing with robust vision-based training," *IEEE transactions on robotics*, vol. 27, no. 3, pp. 545–557, 2011.

[32] N. Ueda, S.-i. Hirai, and H. T. Tanaka, "Extracting rheological properties of deformable objects with haptic vision," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3902–3907.

[33] N. N. A. Charniya and S. V. Dudul, "Sensor for classification of material type and its surface properties using radial basis networks," *Sensors Journal, IEEE*, vol. 8, no. 12, pp. 1981–1991, 2008.

[34] H. Zheng, L. Fang, M. Ji, M. Strese, Y. Ozer, and E. Steinbach, "Deep learning for surface material classification using haptic and visual information," *arXiv preprint arXiv:1512.06658*, 2015.

[35] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," *arXiv preprint arXiv:1511.06065*, 2015.

[36] M. A. Schaeffer and A. M. Okamura, "Methods for intelligent localization and mapping during haptic exploration," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 4. IEEE, 2003, pp. 3438–3445.

[37] N. Alt and E. Steinbach, "Navigation and manipulation planning using a visuo-haptic sensor on a mobile platform," 2014.

[38] C. Fox, M. Evans, M. Pearson, and T. Prescott, "Tactile slam with a biomimetic whiskered robot," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4925–4930.

[39] T. Bhattacharjee, P. M. Grice, A. Kapusta, M. D. Killpack, D. Park, and C. C. Kemp, "A robotic system for reaching in dense clutter that integrates model predictive control, learning, haptic mapping, and planning," in *Proceedings of the 3rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Robots in Clutter: Perception and Interaction in Clutter*, 2014.

[40] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 746–760.

[41] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1817–1824.

[42] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3d object dataset: Putting the kinect to work," in *Consumer Depth Cameras for Computer Vision*. Springer, 2013, pp. 141–165.

[43] Q. Zhang, X. Song, X. Shao, R. Shibasaki, and H. Zhao, "Category modeling from just a single labeling: Use depth information to guide the learning of 2d models," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 193–200.

[44] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Advances in Neural Information Processing Systems*, 2011, pp. 244–252.

[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[46] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014. [Online]. Available: http://dx.doi.org/10.7717/peerj.453

[47] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[48] G. Bradski, *Dr. Dobb's Journal of Software Tools*.

[49] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[50] A. Ciptadi, T. Hermans, and J. M. Rehg, "An In Depth View of Saliency," in *British Machine Vision Conference (BMVC)*, September 2013.

[51] A. Richtsfeld, "The Object Segmentation Database (OSD)," http://http://www.acin.tuwien.ac.at/?id=289, 2012.

[52] T. Bhattacharjee, A. Jain, S. Vaish, M. D. Killpack, and C. C. Kemp, "Tactile sensing over articulated joints with stretchable sensors," in *World Haptics Conference (WHC), 2013*. IEEE, 2013, pp. 103–108.

[53] T. Bhattacharjee, A. Kapusta, J. M. Rehg, and C. C. Kemp, "Rapid categorization of object properties from incidental contact with a tactile sensing robot arm," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 219–226.

[54] "ROS Fuerte," http://wiki.ros.org/fuerte.

[55] "ROS package to convert between ROS and OpenCV Images," http://wiki.ros.org/cv_bridge/Tutorials/UsingCvBridgeToConvertBetweenROSImagesAndOpenCVImages.

[56] "General Hidden Markov Model Library," http://ghmm.org/.

[57] M. D. Killpack, A. Kapusta, and C. C. Kemp, "Model predictive control for fast reaching in clutter," *Autonomous Robots*, pp. 1–24, 2015.