Combining Tactile Sensing and Vision for Rapid Haptic Mapping

Tapomayukh Bhattacharjee, Ashwin A. Shenoi, Daehyung Park, James M. Rehg, and Charles C. Kemp

Abstract— We consider the problem of enabling a robot to efficiently obtain a dense haptic map of its visible surroundings using the complementary properties of vision and tactile sensing. Our approach assumes that visible surfaces that look similar to one another are likely to have similar haptic properties. We present an iterative algorithm that enables a robot to infer dense haptic labels across visible surfaces when given a color-plus-depth (RGB-D) image along with a sequence of sparse haptic labels representative of what could be obtained via tactile sensing. Our method uses a color-based similarity measure and connected components on color and depth data. We evaluated our method using several publicly available RGB-D image datasets with indoor cluttered scenes pertinent to robot manipulation. We analyzed the effects of algorithm parameters and environment variation, specifically the level of clutter and the type of setting, like a shelf, table top, or sink area. In these trials, the visible surface for each object consisted of an average of 8602 pixels, and we provided the algorithm with a sequence of haptically-labeled pixels up to a maximum of 40 times the number of objects in the image. On average, our algorithm correctly assigned haptic labels to 76.02% of all of the object pixels in the image given this full sequence of labels. We also performed experiments with the humanoid robot DARCI reaching in a cluttered foliage environment while using our algorithm to create a haptic map. Doing so enabled the robot to reach goal locations using a single plan after a single greedy reach, while our previous tactile-only mapping method required 5 or more plans to reach each goal.

I. INTRODUCTION

Tactile sensing can provide distinctive information about a robot's environment, including direct measurements of mechanical properties of nearby objects, but requires that the robot make physical contact with locations of interest, which can be energetically costly and time consuming. Vision can rapidly sense visible surfaces, even when they are out of reach, but provides information that differs from what tactile sensing can provide. In this paper, we address the problem of inferring dense haptic labels across a visible surface based on sparse haptic labels, thereby enabling a robot to efficiently obtain a dense map with fewer contact events. One notable characteristic of this problem is the different forms of spatial sampling associated with these two perceptual modalities. We define a haptic map to be a set of pairs associating locations with haptic labels. Specifically, we use two types of haptic map in this paper. For one, the robot must assign haptic labels to point cloud data, specifically from an RGB-D image. For the other, the robot must assign haptic labels
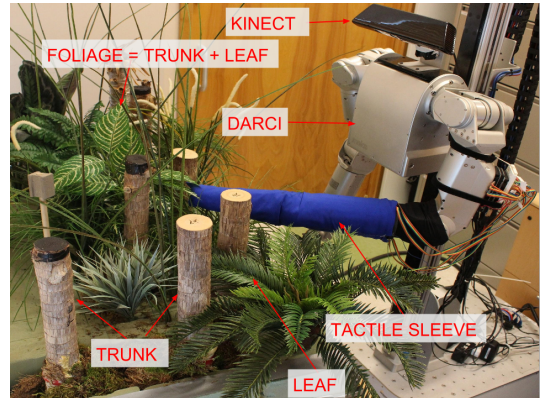
T. Bhattacharjee, A. A. Shenoi, D. Park, and C. C. Kemp are with the Healthcare Robotics Lab, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology,

J. M. Rehg is with the Computational Perception Lab, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology,

*T. Bhattacharjee is the corresponding author {tapomayukh@gatech.edu}.

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

to occupied voxels. The goal is for the robot to rapidly and accurately assign haptic labels to support manipulation. We refer to the process of the robot assigning haptic labels to locations as haptic mapping.

Our key idea for this work is that visually similar objects near the robot are likely to have similar haptic properties. Though touch typically provides information about a small area, humans can attribute a haptic property to a large area by visually perceiving that the large area looks like the small contacted area. Humans can also attribute haptic properties to objects based on visually similar objects with which they have interacted in the past (e.g. fabric identification) [1].

We performed simulations to analyze the performance of our algorithm on different cluttered environments (See Section IV). We discuss the effects of algorithm parameters and present the results with our method in Section IV-A. As shown in Figure 1, we also performed experiments with the humanoid robot DARCI reaching in a cluttered environment using a tactile-sensing sleeve and a Kinect (Section V and Section V-D). These experiments build on our past research devoted to reaching in clutter and use the same reconfigurable artificial foliage and underlying system components, including a tactile perception system that recognizes object categories based on touch [2]. For our experiments with the real robot, this tactile perception system provided sparse haptic labels of trunk and leaf.

II. RELATED WORK

Researchers have worked on various aspects of the role of multiple sensory modalities such as haptics and vision in day-to-day manipulation tasks for both humans and robots.

Research on multisensor fusion has often focused on combining overlapping information to obtain a reliable estimate of the environment [3], and other work has focused on using modalities sequentially to guide each other, such as vision providing guidance for tactile exploration [4]. Our work considers the problem of using sparse tactile sensing with dense visual sensing to produce dense haptic representations of the robot's environment.

A. Human Multisensory Perception

Many studies on human multisensory perception focus on cognitive and neurological aspects, such as the binding problem [5] in cross-modal interactions [6]. Others have investigated the psychophysical aspects of combined perception with vision and haptics [7]–[9] using the concept of temporal synchrony and spatial coincidence. Our method uses spatial and temporal correspondence between haptic and visual sensory modalities to obtain a dense haptic map.

B. Robotic Multisensory Perception

1) Object Perception: There are many studies that deal with haptic and visual perception of virtual objects [10]–[12] or real remote objects [13] by a human operator. Researchers have also investigated haptic and visual perception to support object perception for robot autonomy. Stansfield [4] used vision to segment an object and estimate its position and then used haptics to actively explore and perceive the object. Allen [14] used vision to obtain sparse 3-D data about regions of interest and then used haptics to actively explore regions for object recognition. Researchers have also extracted object attributes such as rheological properties [15], mass and elasticity [16], friction [17], and such by exerting known forces with haptics and observing responses with vision.

2) Active Perception: Researchers have worked on active vision and associated information from vision to various physical interactions [18]–[21]. Coelho et al. [22] used vision to determine appropriate grasping strategies and then used haptics to grasp the objects. Using tasks such as flipping a light switch and operating a drawer, Nguyen and Kemp implemented SVMs with vision as input to predict if a manipulation behavior is likely to succeed at a particular 3D location [23]. Sukhoy and Stoytchev developed a framework for pressing buttons using a robot's visual and auditory percepts [24]. van Hoof et al. used vision in cluttered environments to predict an action with the highest information gain [25].

3) Localization and Mapping: Relatively few studies have looked at mapping using both haptics and vision modalities. However, researchers have studied various ways of haptically mapping the environment by assigning physical properties to objects using only tactile sensors. Some studies focused on presenting haptic information using a 'haptograph' that uses frequency and spatial analysis to represent contact information [26], [27]. Schaeffer and Okamura [28] used various probabilistic methods to simultaneously localize the movement of a robotic fingertip while haptically mapping the surface. Rui et al. used Gelsight tactile sensing to localize the pose of small parts grasped using a robot hand [29]. Alt and

---

Algorithm 1 VisualHapticRelation(SH_Labels, RGB_Im)

Input: $SH\_Labels \leftarrow Sparse\ Haptic\ Labels$
Input: $RGB\_Im \leftarrow RGB\ Image\ from\ Kinect$
Output: $CL\_list \leftarrow Color\ Label\ relation\ list$
Output: $LCL\_list \leftarrow Location\ Color\ Label\ relation\ list$
5: **while** SH_Labels **is not empty do**
$\quad SH \leftarrow SH\_Labels.POP()$
$\quad XY \leftarrow SH.XY$
$\quad RGB \leftarrow RGB\_Im[XY]$
$\quad Label \leftarrow SH.Haptic\_Label$
10: $\quad LCL \leftarrow LCL\_list$ **entry with best match** XY
$\quad$ **if** $LCL == None$ **then**
$\quad\quad Label\_counts \leftarrow$ **new 0 array**
$\quad\quad Label\_counts[Label] + +$
$\quad\quad LCL \leftarrow (XY, RGB, Label)$
15: $\quad\quad LCL\_list.Append(LCL)$
$\quad$ **else**
$\quad\quad LCL.Label\_counts[Label] + +$
$\quad$ **end if**
$\quad CL \leftarrow CL\_list$ **entry with best match** RGB
20: $\quad$ **if** $CL == None$ **then**
$\quad\quad Label\_counts \leftarrow$ **new 0 array**
$\quad\quad Label\_counts[Label] + +$
$\quad\quad CL \leftarrow (RGB, Label\_counts)$
$\quad\quad CL\_list.Append(CL)$
25: $\quad$ **else**
$\quad\quad CL.Label\_counts[Label] + +$
$\quad$ **end if**
end while

Steinbach developed a visuo-haptic sensor which uses vision to monitor the deformation of a plastic foam in contact to attribute haptic properties to objects in the environment [30]. Fox et al. [31] used data from a whiskered robot for grid-based Tactile SLAM to generate an occupancy grid. All these studies focus on building the haptic map based on tactile data alone, and thus, the maps are local and limited to the area of active exploration.

Unlike the above active exploration studies, our previous work [2], [32] uses information generated from *incidental contact* with a whole-arm tactile sensor to generate a local and sparse 3D haptic map. By incidental contact, we mean contact that is not central to the robot's current actions and may occur unexpectedly or unintentionally. In this paper, our system uses vision to make better use of information gained via incidental contact. For example, the sparsity of the haptic maps used in our previous work can sometimes reduce the usefulness of plans [32]. Our current study addresses this issue by using vision to generate dense haptic maps across visible surfaces based on sparse haptic labels.

III. ALGORITHM

Our proposed algorithm is divided into 2 stages, assigning sparse haptic labels (See Section III-A) and assigning dense haptic labels (See Section III-B). For this paper, our algorithm makes two notable simplifications. First, it only uses color to decide which locations are visually similar. Additional appearance features, such as texture features, could potentially improve its performance. Second, it only uses haptic labels with 2D image coordinates and performs operations with respect to the image's coordinate system.

Algorithm 2 Map(RGB_Im,D_Im,CL_list,LCL_list)

    Input: $RGB\_Im \leftarrow RGB\ Image\ from\ Kinect$
    Input: $D\_Im \leftarrow Depth\ Image\ from\ Kinect$
    Input: $CL\_list \leftarrow Color\ Label\ relation\ list$
    Input: $LCL\_list \leftarrow Location\ Color\ Label\ relation\ list$
5: **Output:** $Hap\_map \leftarrow Haptic\ map\ of\ visible\ scene$
    procedure GETLABEL(RGB, CL_list)
        $CL \leftarrow CL\_list$ **entry with best match** RGB
        if $CL == None$ **then**
            $Label \leftarrow$ "Unclassified"
10:     **else**
            $Label \leftarrow ArgMax\,(CL.Label\_counts)$
            $Count \leftarrow CL.Label\_counts\,[Label]$
            if $Count \leq 0.8 * Sum\,(CL.Label\_counts)$ **then**
                $Label \leftarrow$ "Uncertain"
15:        **end if**
        end if
        return $Label$
    end procedure
    for each $Pixel$ **in** RGB_Im **do**
20:     $RGB \leftarrow Pixel.RGB$
        $Pixel.Label \leftarrow GetLabel\,(RGB, CL\_list)$
    end for
    for each LCL **in** LCL_list **do**
        $RGB \leftarrow LCL.RGB$
25:     $Label \leftarrow GetLabel\,(RGB, CL\_list)$
        if $LCL.Label \neq Label$ **then**
            $XY \leftarrow LCL.XY$
            $C\_RGB \leftarrow ConnectedComponent(RGB\_Im, XY)$
            $C\_D \leftarrow ConnectedComponent(D\_Im, XY)$
30:       $Segment \leftarrow C\_RGB \cap C\_D$
            for each $Pixel$ **in** $Segment$ **do**
                $Pixel.Label \leftarrow LCL.Label$
            end for
        end if
35: **end for**
    for each $Pixel$ **in** RGB_Im **do**
        $Hap\_map.Add\,(Pixel)$
    end for

Tactile sensors can assign haptic labels to 3D Cartesian locations around a robot using forward kinematics with calibrated sensors, and our algorithm assumes that these have been transformed into the 2D color image.

A. Stage 1: Sparse Haptic Labeling

Our system on the real robot uses our previous hidden Markov model (HMM) based classification method described in [2] for local haptic categorization. It takes force data from the tactile-sensing sleeve over time as input and outputs sparse haptic labels each with a 2D color image coordinate.

B. Stage 2: Dense Haptic Labeling

In this stage, our algorithm uses the sparse haptic labels from Stage 1 and RGB-D data from a Kinect to assign dense haptic labels across visible surfaces. In order to infer the dense haptic labels across visible surfaces from the sparse haptic labels, our algorithm maintains a visual-haptic relation.

1) Visual-Haptic Relation: The algorithm (see Algorithm 1) creates a relation between the visual and haptic data using two lists.

The first list (LCL_list) is the 'Location Color Label' relation, which keeps track of the locations of all the points of contact, their corresponding RGB values and haptic labels assigned in Stage 1. When a new contact is made, the algorithm finds the corresponding point in the image. It checks if the new point is the same as any of the previously tracked points. If such a point exists, it increments the corresponding haptic label. Otherwise, it stores the coordinates of this new point, its RGB value, and haptic label counts in the list.

The second list (CL_list) is the '*Color Label*' relation between colors and haptic labels. When a new contact is made, the algorithm compares the RGB value of this point and checks if the color is similar (We use a distance measure in the CIELAB color space, see Section IV) to any of the colors of previously tracked points. If such a point exists, the algorithm increments the count of the corresponding haptic label for this color. Otherwise, it creates a new relation for this color. The algorithm uses this relation in Section III-B.2.

2) Dense Haptic Map Generation: In this stage, our algorithm (see Algorithm 2) uses CL_list (See Section III-B.1) to infer the haptic labels of the rest of the visible scene. For this, the algorithm compares the color of every point in the visible scene with the colors maintained in the CL_list (using a distance measure in the CIELAB color space, See Section IV). The algorithm determines the appropriate haptic label by finding a label that has a count greater than 80% of the total haptic count for the best matching color, if there is a matching color. If such a label doesn't exist, then the point is classified as 'Uncertain'. Any points in the visible scene that do not match a color maintained in the CL_list remain 'Unclassified'.

However, there may be scenarios in which objects with visually similar properties have distinct haptic labels. The algorithm detects such cases using contradictions between CL_list and LCL_list. For example, a new contact could be made and the haptic label for the color associated with the point (obtained from LCL_list) could be different from the haptic label for the color in general (obtained from CL_list). The algorithm addresses such situations by updating only a local segmented region (instead of the whole scene) with the associated haptic label. The algorithm segments a region by computing connected components for the RGB image, computing connected components for the depth image, selecting the color and depth connected components that contain the point of interest (obtained from LCL_list), and then finding the intersection between these two connected components. Section IV-A.2 shows the relevant quantitative results.

3) Implementation: We implemented our algorithm in Python using the scikit-image [33], OpenCV [34] and NumPy [35] libraries. We used the scikit-image Python library for color transforms and color difference calculations, OpenCV for image handling and NumPy for data manipulation. We used 'cv_bridge' [36] to convert between ROS images and OpenCV images. After we updated the RGB image with dense haptic labels, we combined the RGB image with the depth image from the Kinect to generate a point cloud using nodelets in the depth_image_proc ROS package [37] to assign haptic labels to each point in the point cloud. The implementation we evaluated for this paper directly transforms haptic labels in 3D to the image's coordinate

TABLE I: Effect of threshold and distance measure on performance.

Distance Measure	Threshold	$F_1 score$ [0, 1]	
		(Avg.)	(Std.Dev.)
CIE76	5	0.55	0.18
	15	0.75	0.17
	25	0.75	0.18
CIE94	5	0.57	0.23
	15	0.63	0.24
	25	0.66	0.23
CIEDE2000	5	0.62	0.17
	15	**0.76**	**0.17**
	25	0.72	0.21

TABLE II: Percentage of pixels updated with contacts for all images.

ContactPoints/ No.ofObjects	Pixels updated correctly	
	(Avg.)	(Std.Dev)
5.0	63.08 %	19.71 %
10.0	69.48 %	18.26 %
15.0	71.98 %	17.28 %
20.0	73.84 %	17.02 %
25.0	75.11 %	16.18 %
30.0	75.67 %	16.09 %
35.0	74.98 %	16.91 %
40.0	**76.02 %**	**16.26 %**

system, ignoring their depth with respect to the visible surface. Filtering out haptic labels that differ too much in depth from the visible surface would be a worthwhile modification.

IV. Evaluation with Simulated Trials

We evaluated our algorithm on annotated RGB-D images in order to provide a controlled evaluation with a substantial number of trials. For this evaluation, we used a simple model of a robot, which provided a sequence of haptic labels with each label associated with a specific pixel in the RGB-D image.

To create our dataset for this evaluation, we selected 186 RGB-D images of indoor cluttered scenes suitable for robot manipulation tasks from various publicly available RGB-D datasets [38]–[42]. We relabeled the segmented objects in these images using tools from [43], applying a single haptic label to each segmented object and, hence, all of its pixels. The haptic labels we assigned were $Books, Cardboard, Ceramic, Fabric, Foam, Glass, Leather, Metal, Onion, Paper, Plastic, Rubber, Sponge, Wax, Wood, Bread, Plant, Soap, Cinder Block$ and $Soft Plastic$. We chose these haptic labels because tactile sensing could plausibly make these distinctions. Force, deformation, area of contact, texture, stiffness, heat transfer and other haptic features have been used to make comparable distinctions in prior research [44]–[49]. Three people independently assigned these haptic labels to the segmented objects in the 186 RGB-D images. When the haptic labels for an object disagreed, the three people discussed the label and attempted to come to a consensus. For the fewer than 5 objects for which consensus was not readily achieved, they found real-world objects that matched the objects in the images and physically interacted with them to achieve consensus. One experimenter also categorized the images

based on scenes (table top, shelf, sink area, bed, floor and misc.) and clutter density (low and high).

We performed two sets of simulations with the 186 RGB-D images. For the first set of simulations, we analyzed the performance with different distance measures and thresholds. In the first set of simulations, we randomly selected 40 labeled pixels for each image from the pixels associated with the segmented and labeled objects. We evaluated the generated dense haptic map matching it against the ground truth labels. We repeated this process 5 times for each image and each parameter (3 color-distance measures, and 3 color-distance thresholds, See Table I) to generate $5 * 3 * 3 * 40 * 186 = 334800$ trials. For each parameter, we computed the average across all the images for each category. We used the $F_1 score$ to evaluate the system for different thresholds and distance measures.

We selected the 3 distance measures CIE76, CIE94, and CIEDE2000 according to [50], [51]. CIE76 (1976) was the original distance metric developed for L*a*b space which was based on Euclidean distance but had issues with perceptual uniformity in saturated regions [50], [52]. The revised version in 1994, CIE94, was defined in L*c*h color space and introduced application-specific weights in the formula but still was not perceptually uniform [50], [52]. CIEDE2000 added compensation terms in the CIE94 formula to revise it [51], [52]. Section IV-A.2 discusses the results.

For the second set of simulations, we used the color-distance measure and color-distance threshold that gave the best performance in our first tests. 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.

A. Results from the Simulated Trials

1) First Set - Effect of Algorithm Parameters: We evaluated the algorithm for different color-distance measures, and color-distance threshold values using all the images in the dataset. Table I shows the results for different values of the parameters. Our algorithm performed best using CIEDE2000 with '15' as the threshold, so we chose CIEDE2000 as the color-distance metric for the second set of simulations and the robot experiments described in Section V.

2) Second Set - Quantitative Evaluation of Algorithm Performance: 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. Table II shows the results. As the number of contacts increased, the rate at which the pixels were correctly updated decreased. Although our sampling method tended to distribute the contact points across the objects, it was still highly random. 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
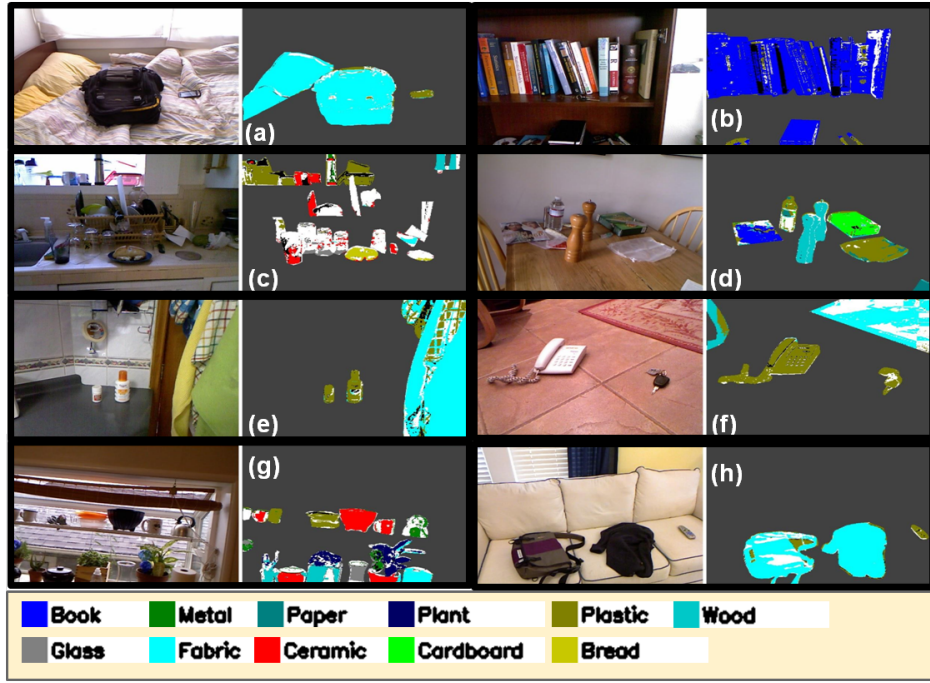
Fig. 2: *Simulation results of haptic categorization for some example images from different publicly available datasets [38]–[42]. These examples show scenes from different environments with varying density of clutter. The uncertain regions are marked with white and the background is marked with grey.*

TABLE III: Performance on different environments.

Env. Type	$F_1 score$ [0, 1]		Pixels updated correctly	
	(Avg.)	(Std.Dev.)	(Avg.)	(Std.Dev.)
Low Clutter	**0.84**	**0.16**	**79.32%**	**16.12%**
High Clutter	0.72	0.20	70.10%	14.74%
Bed	0.78	0.09	77.61%	21.23%
Floor	**0.92**	**0.03**	**92.56%**	**7.82%**
Shelf	0.71	0.15	68.08 %	7.50%
Sink Area	0.79	0.18	68.26%	13.38%
Table Top	0.80	0.20	79.01%	15.50%
Misc.	0.88	0.13	78.21%	24.70%

per object, the algorithm correctly updated an average of 76.02% 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.

3) Second Set - 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. 3 show the results. Our algorithm performed better with low-clutter environments ($F_1 score = 0.84$) when compared to high-clutter environments ($F_1 score = 0.72$).

4) Second Set - Effect of Type of Environment: We also classified the images into 6 different scene-based categories as described in Section IV. We computed the same performance measurements as in Section IV-A.3. Table III and Fig. 4 show the results.

V. EVALUATION WITH A REAL ROBOT

A. Experimental Apparatus

1) The Robot: We used the humanoid robot DARCI, 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 left arm had a 3D-printed cylindrical ABS plastic end effector (Fig. 1). 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.

2) The Tactile Sensor: For tactile sensing, we used our fabric-based tactile-sensing sleeve with 25 discrete tactile sensing areas (taxels) on the robot's forearm and end effector [53]. The tactile perceptual system converted the raw taxel measurements to approximate normal forces using a non-linear calibration function.

3) The Vision Sensor: For the visual modality, we used a Microsoft Kinect to capture color (RGB) and depth images of the scene. Our algorithm processes the RGB and depth data as described in Section III-B.

4) The Environment: We evaluated our system in a cluttered environment consisting of artificial foliage that we have used in our previous research [32]. Figure 1 shows the environment, which is composed of trunks and leaves.

B. Experimental Procedure

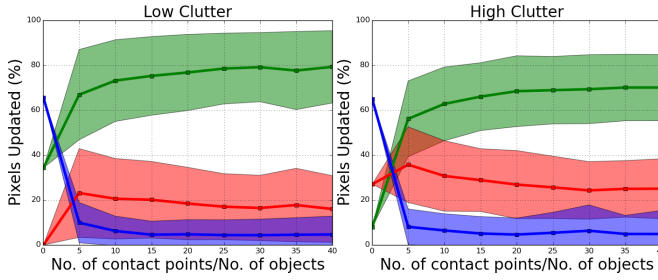We used a system that runs Ubuntu 12.04 32-bit OS with a 3.5.0-51-generic-pae linux kernel. It has 16 GB RAM and an

Fig. 3: *Percentage of pixels updated for environments with different clutter densities. Green: Correct, Red: Incorrect/Uncertain, Blue: Unclassified*
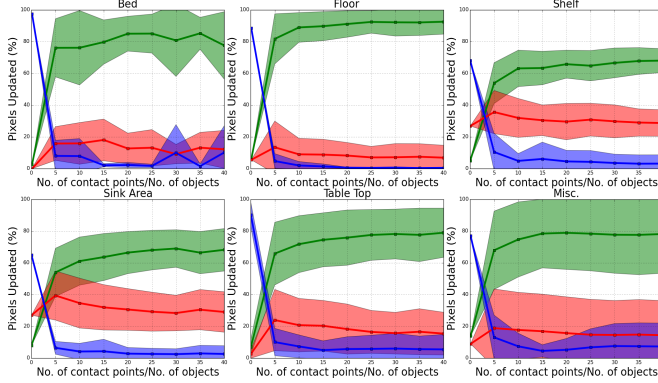

Fig. 4: *Percentage of pixels updated for environments categorized based on scene. Green: Correct, Red: Incorrect/Uncertain, Blue: Unclassified*

Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz X 8 processor. We used ROS Fuerte [54] for communicating with the RTPC and the robot DARCI. In our experiment, the robot reached to different goal locations in the foliage environment, while making incidental contact with various objects (trunk and leaf). Our algorithm used the haptic classification of contact into trunk or leaf along with the RGB-D data from the Kinect to infer a dense haptic map across a visible surface.

To demonstrate our algorithm, we first teleoperated the robot using interactive markers in rviz, a ROS-based 3D visualization tool, resulting in various trajectories, contact events, and lighting conditions. The robot used our previously developed controller described in [32] to quickly reach to a target location with low contact forces .

For our quantitative evaluation using autonomous reaching, we selected 1 start location and 2 goal locations within the robot arm's reach and inside the Kinect's field of view (See Fig. 6). When reaching into the clutter, the robot only used the joint-space controller to follow joint-space trajectories as described in [32], not the task-space controller. For all reaches, the robot first reached from the start location to the goal location when there was no map available. At any time, if the robot's arm became blocked, it moved back to the start location and planned a new joint-space trajectory using the current haptic map. At all times, the robot was updating its haptic map, which was erased prior to the start of each trial. In total, we conducted 12 trials. 6 trials (3 reaches

to 2 goal locations) tested reaching when the robot used vision, tactile sensing, and our algorithm to generate dense haptic maps, and 6 trials (3 reaches to 2 goal locations) tested reaching when the robot only used tactile sensing to generate sparse haptic maps. Section V-D discusses the results.

C. Data Collection and Preprocessing

1) The Tactile Sensor: We recorded the force and Cartesian 3D position data from each taxel of the tactile-sensing sleeve at 100 Hz, truncated the data to begin at the estimated onset of contact, and fed the force data into the trained HMM models for haptic category predictions.

2) The Vision Sensor: For the visual data, we collected RGB-D data from the Kinect. We transformed the color space from RGB to CIELAB space. This color model is designed to approximate the human vision system [55]. We also collected the depth image from the Kinect. Since the depth data received has missing data, we performed a Naiver-stokes-based method for 'inpainting' from OpenCV [34] in order to fill missing data in the depth image.

D. Results from the Real Robot

1) Demonstrations: Figure 5 shows the dense haptic map generated for some of the teleoperated demonstration trials. Note that the unclassified region also includes the area occupied by the robot-arm.

2) Quantitative Evaluation: Fig. 7 shows the results of our quantitative evaluation. Using our method for obtaining a dense map by combining vision and tactile sensing, the joint-space planner was able to reach both the goal locations in all the trials using just one plan. In contrast, when not using vision, the robot failed two times (greater than 10 plans) and took 5 or more plans for trials in which it succeeded. This evaluation demonstrates that rapidly generating a dense haptic map can be useful. However, using our task-space planner for greedy reaching as in [32] could potentially have improved performance dramatically. Fig. 8 shows an example plan of the robot with and without our new algorithm. Since the goal of the planner is to avoid rigid objects, the haptic map only updated the rigid objects (i.e., trunks) using 3D voxels.

VI. CONCLUSIONS

We presented a method to more efficiently obtain dense haptic maps across visible surfaces using sparse haptic labels provided by tactile sensing. 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 analyze the performance of our algorithm, we simulated haptic contact and applied our algorithm to a collection of 186 indoor cluttered images pertinent to robot manipulation selected from various publicly available RGB-D datasets [38]–[42]. We discussed the effect of algorithm parameters and environment types on the performance. With 40 contact points per object out of an average 8602 contact points per object for all images, the algorithm correctly updated 76.02% of the pixels in the images. The algorithm can also reach an average $F_1 score$ of 0.84 for low-cluttered environments
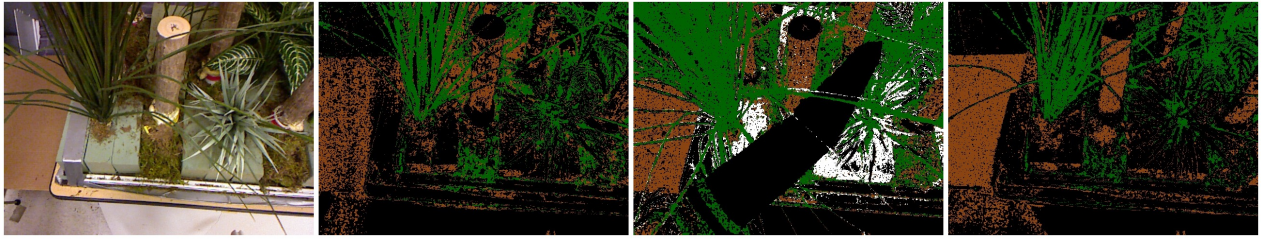
Fig. 5: *Dense haptic map generated during various demonstration trials. Legend: Brown: Trunk, Green: Leaf, White:Uncertain, Black: Unclassified. The algorithm updated the haptic map in real time with 25 Hz. frequency.*
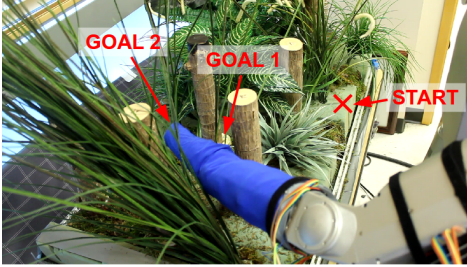

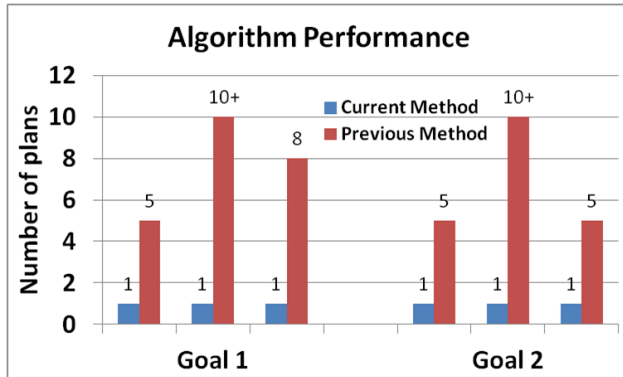Fig. 6: *One start and two goal locations. A robot reaches through the clutter to 'Goal 2' location.*

Fig. 7: *Number of plans required to reach the different goal locations using our current method and previous method of haptic mapping [2].*

and 0.92 for floor scenes. It performs better for low-clutter scenes than for high-clutter scenes. As expected, with more contacts our algorithm performs better at inferring the correct haptic labels for the environment. We also conducted real robot experiments with a tactile-sensing sleeve and a Kinect under various conditions. Our results demonstrated that our algorithm can rapidly create dense haptic maps useful for reaching goals in clutter.
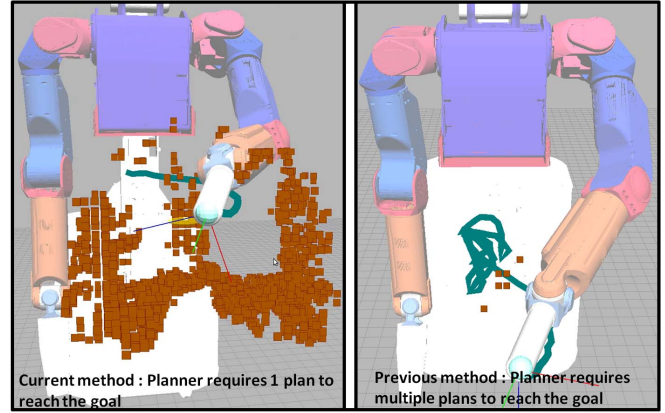
ACKNOWLEDGMENT

Fig. 8: *Left: Robot reaching the target with only one plan (one blue line) with current method. Right: Multiple plans required to reach the goal with previous method [2]. Blue lines show end-effector trajectories. Only rigid obstacles are mapped using 3D voxels.*

REFERENCES

[1] Z. Xue, "Study on relations between visual and haptic perceptions of textile products," Ph.D. dissertation, Lille 1, 2012.

[2] 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.

[3] J. K. Hackett and M. Shah, "Multi-sensor fusion: a perspective," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. IEEE, 1990, pp. 1324–1330.

[4] 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.

[5] A. Treisman, "The binding problem," *Current opinion in neurobiology*, vol. 6, no. 2, pp. 171–178, 1996.

[6] S. Gepshtein, J. Burge, M. O. Ernst, and M. S. Banks, "The combination of vision and touch depends on spatial proximity," *Journal of Vision*, vol. 5, no. 11, p. 7, 2005.

[7] R. L. Klatzky, S. J. Lederman, and C. Reed, "There's more to touch than meets the eye: The salience of object attributes for haptics with and without vision." *Journal of experimental psychology: general*, vol. 116, no. 4, p. 356, 1987.

[8] B. Jones and S. ONeil, "Combining vision and touch in texture perception," *Perception & Psychophysics*, vol. 37, no. 1, pp. 66–72, 1985.

[9] H. B. Helbig and M. O. Ernst, "Knowledge about a common source can promote visual-haptic integration," *Perception-London*, vol. 36, no. 10, pp. 1523–1534, 2007.

[10] J. K. Salisbury and M. A. Srinivasan, "Phantom-based haptic interaction with virtual objects," *Computer Graphics and Applications, IEEE*, vol. 17, no. 5, pp. 6–10, 1997.

[11] A. Lécuyer, "Simulating haptic feedback using vision: A survey of research and applications of pseudo-haptic feedback," *Presence:*

Teleoperators and Virtual Environments, vol. 18, no. 1, pp. 39–53, 2009.

[12] Y. Yokokohji, R. L. Hollis, and T. Kanade, "What you can see is what you can feel-development of a visual/haptic interface to virtual environment," in *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*. IEEE, 1996, pp. 46–53.

[13] C. W. Kennedy, T. Hu, J. P. Desai, A. S. Wechsler, and J. Y. Kresh, "A novel approach to robotic cardiac surgery using haptics and vision," *Cardiovascular Engineering: An International Journal*, vol. 2, no. 1, pp. 15–22, 2002.

[14] 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.

[15] N. Ueda, S. Hirai, and H. T. Tanaka, "Extracting rheological properties of deformable objects with haptic vision," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, April 2004.

[16] H. T. Tanaka, K. Kushihama, N. Ueda, and S. Hirai, "A vision-based haptic exploration," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 3441–3448.

[17] D. Yamashiro, S. Tanaka, and H. T. Tanaka, "Active estimation of friction properties with haptic vision," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*. IEEE, 2008, pp. 1329–1332.

[18] L. Berthouze, P. Bakker, and Y. Kuniyoshi, "Learning of oculo-motor control: a prelude to robotic imitation," in *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1996, pp. 376–381.

[19] L. Berthouze and Y. Kuniyoshi, "Emergence and categorization of coordinated visual behavior through embodied interaction," *Autonomous Robots*, vol. 5, no. 3-4, pp. 369–379, 1998.

[20] M. HUlse, S. McBride, and M. Lee, "Developmental robotics architecture for active vision and reaching," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–6.

[21] N. J. Butko and J. R. Movellan, "Learning to look," in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*. IEEE, 2010, pp. 70–75.

[22] J. Coelho, J. Piater, and R. Grupen, "Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot," *Robotics and Autonomous Systems*, vol. 37, no. 2, pp. 195–218, 2001.

[23] H. Nguyen and C. C. Kemp, "Autonomously learning to visually detect where manipulation will succeed," *Autonomous Robots*, vol. 36, no. 1-2, pp. 137–152, 2014.

[24] V. Sukhoy and A. Stoytchev, "Learning to detect the functional components of doorbell buttons using active exploration and multimodal correlation," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, 2010, pp. 572–579.

[25] H. Van Hoof, O. Kroemer, H. Ben Amor, and J. Peters, "Maximally informative interaction learning for scene exploration," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5152–5158.

[26] Y. Yokokura, S. Katsura, and K. Ohishi, "Haptic recognition and mapping of driving road environment by haptograph," in *SICE, 2007 Annual Conference*. IEEE, 2007, pp. 2296–2301.

[27] S. Katsura, Y. Yokokura, and K. Ohishi, "Acquisition and visualization of personal characteristics by haptograph," in *Advanced Motion Control, 2008. AMC'08. 10th IEEE International Workshop on*. IEEE, 2008, pp. 434–439.

[28] 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.

[29] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, "Localization and manipulation of small parts using gelsight tactile sensing," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3988–3993.

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

[31] 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.

[32] 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.

[33] 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

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

[35] 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.

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

[37] "ROS Package for Depth Image Processing," http://wiki.ros.org/depth_image_proc.

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

[39] 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.

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

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

[43] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[44] 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.

[45] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. Martinez Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell, *et al.*, "Using robotic exploratory procedures to learn the meaning of haptic adjectives," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3048–3055.

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

[47] S. Takamuku, G. Gomez, K. Hosoda, and R. Pfeifer, "Haptic discrimination of material properties by a robotic hand," in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*. IEEE, 2007, pp. 1–6.

[48] N. Jamali and C. Sammut, "Majority voting: material classification by tactile sensing using surface texture," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 508–521, 2011.

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

[50] A. R. Robertson, "Historical development of cie recommended color difference equations," *Color Research & Application*, vol. 15, no. 3, pp. 167–170, 1990.

[51] M. R. Luo, G. Cui, and B. Rigg, "The development of the cie 2000 colour-difference formula: Ciede2000," *Color Research & Application*, vol. 26, no. 5, pp. 340–350, 2001.

[52] "Metrics for Color Difference," http://en.wikipedia.org/wiki/Color_difference.

[53] 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.

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

[55] "Lab Color Space," http://en.wikipedia.org/wiki/Lab_color_space.