A CRF THAT COMBINES TACTILE SENSING AND VISION FOR HAPTIC MAPPING

A Thesis
Presented to
The Academic Faculty

by

Ashwin A. Shenoi

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2016

A CRF THAT COMBINES TACTILE SENSING AND VISION FOR HAPTIC MAPPING

Approved by:

Professor Charles C. Kemp, Advisor
Wallace H. Coulter Department of Biomedical Engineering
Georgia Institute of Technology

Professor Patricio Antonio Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor James Hays
School of Interactive Computing, College of Computing
Georgia Institute of Technology

Date Approved: 29 April 2016

ACKNOWLEDGEMENTS

I thank my advisor, Dr Kemp, for giving me the right guidance to complete this work. I thank Dr Hays and Dr Vela for agreeing to be part my thesis reading committee. I am grateful to Tapo for mentoring me. This work would not have been possible without his guidance. I thank Ari, Daehyung, Yash and Phil, for helping me out at various points of time during my time at Healthcare Robotics Lab. I thank my friends Aamedh, Adi, Guru and Rohan for the support they provided during the last 2 years. I am grateful to my family for their love and support. Lastly, I thank Archana for being so supportive.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . . **iii**

LIST OF TABLES . . . . . **vi**

LIST OF FIGURES . . . . . **vii**

LIST OF SYMBOLS OR ABBREVIATIONS . . . . . **viii**

SUMMARY . . . . . **ix**

I INTRODUCTION **1**

II RELATED WORK **3**

2.1 Material recognition using tactile sensing 3

2.2 Material recognition using vision 3

2.3 Integration of vision and touch 4

2.4 Haptic Mapping of the scene 5

III GENERATING A DENSE HAPTIC MAP **7**

3.1 Previous Algorithm 7

3.1.1 Visual-Haptic Relation 7

3.1.2 Dense Haptic Map Generation 8

3.2 Using a dense CRF to generate Dense Haptic Map 10

3.3 Dataset for evaluation 12

3.4 Comparison with our previous algorithm [13] 13

3.4.1 Effect of Clutter 16

3.4.2 Effect of Type of Environment 16

3.5 Implementation: 16

IV MATERIAL RECOGNITION USING VISION **18**

4.1 Material Recognition using CNN 18

4.2 Fine tuning the CNN model 18

4.3 Effect of prior probability p^v 20

4.3.1 Uniform Distribution 24
4.3.2 Prior determined by CNN 24
V EXPLORATION USING ENTROPY MEASURE 25
5.1 Entropy Measure 25
5.2 Evaluation 25
VI EVALUATION WITH A REAL ROBOT 30
6.1 Experimental Setup 30
6.2 Experimental Procedure 31
6.3 Experimental Results 32
VII CONCLUSIONS AND FUTURE WORK 34
7.1 Conclusion 34
7.2 Future work 35
7.2.1 Dynamic Environment 35
7.2.2 Fine tuning the CNN 35
7.2.3 Different features for visual similarity 35
7.2.4 Other modality for material recognition 35
APPENDIX A — CAFFE SETUP FOR FINE TUNING OF CNN 37
REFERENCES 38

LIST OF TABLES

1 Comparison of performance of our current algorithm with our previous algorithm [13]. 14

2 Performance on different environments after 40 contact points per object using current algorithm. 14

3 Effect of prior on performance using current algorithm. 20

4 Effect of sampling method with prior from CNN. 26

5 Effect of sampling method with Uniform prior. 29

6 Performance of the algorithm on the foliage environment using current algorithm. 33

LIST OF FIGURES

1 *Integrating tactile sensing and vision for material recognition. The vision pipeline is processed by a fully convolution network trained on the MINC dataset by Bell et al. [7]. We use a fabric based tactile sensing sleeve and HMMs [11] to classify points of contact based on their haptic property. We combine the probabilities from the two modalities by taking a convex sum inspired by Arnab et al. [6]. A dense CRF is used to predict labels for each pixel from the combined probability map.* . . 11

2 *Percentage of pixels assigned correct/incorrect labels for environments with different clutter densities. Green: Correct, Red: Incorrect.* 15

3 *Percentage of pixels assigned correct/incorrect labels for environments categorized based on scene. Green: Correct, Red: Incorrect.* 15

4 *Simulation results of haptic categorization for some example images from different publicly available datasets [22, 31, 36, 44, 46]. These examples show scenes from different environments with varying density of clutter.* 17

5 *The fully convolution network fine tuned on our image dataset. This GoogLeNet [54] model was originally fine tuned on the MINC dataset by Bell et al. [7].* 19

6 *Effect of prior on performance of the algorithm.* 21

7 *Figure 6 zoomed in for first 20 contact points.* 22

8 *Figure 6 zoomed in for first 100 contact points.* 23

9 *Effect of sampling method on performance of the algorithm with prior from CNN.* 27

10 *Effect of sampling method on performance of the algorithm with uniform prior.* 28

11 *A robot DARCI, equipped with a tactile-sensing sleeve (blue) and a Kinect, reaching into the cluttered foliage environment.* 31

12 *The top row shows the scene after the robot made 10 reaches. The middle row shows the annotated images. 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. The Goal locations are marked with red X's in the top row.* 32

LIST OF SYMBOLS OR ABBREVIATIONS

CNN Convolutional Neural Network. iv, v, 1, 2, 4, 5, 17, 18, 20, 24, 26, 34, 35

CRF Conditional Random Field. iv, vii, ix, 1, 4, 5, 7, 10–12, 17, 18, 24, 34

DARCI Dynamically Adapting Robot for Cooperative Interactions. vii, ix, 30, 31

HMM Hidden Markov Model. vii, 11, 12, 30, 32

MINC Materials in Context Database. vii, 3, 11, 18–20, 24

ROS Robot Operating System. 30

SEAs Series Elastic Actuators. 30

SUMMARY

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. In our previous work, we introduced an iterative algorithm that enabled a robot to infer dense haptic labels across visible surfaces in an RGB-D image when given a sequence of sparse haptic labels. In this work, we describe how dense conditional random fields (CRFs) can be applied to this same problem and present results from evaluating a dense CRF's performance in simulated trials with idealized haptic labels. We evaluated our method using several publicly available RGB-D image datasets with indoor cluttered scenes pertinent to robot manipulation. In these simulated trials, the dense CRF substantially outperformed our previous algorithm by correctly assigning haptic labels to an average of 93% (versus 76% in our previous work) of all object pixels in an image given the highest number of contact points per object. Likewise, the dense CRF correctly assigned haptic labels to an average of 81% (versus 63% in our previous work) of all object pixels in an image given a low number of contact points per object. We compared the performance of dense CRF using uniform prior with a dense CRF using prior obtained from the visible scene using a Fully Convolutional Network trained for visual material recognition. The use of the convolutional network further improves the performance of the algorithm. We also performed experiments with the humanoid robot DARCI reaching in a cluttered foliage environment while using our algorithm to create a haptic map. The algorithm correctly assigned the label to 82.52% of the scenes with trunks and leaves after 10 reaches into the environment.

CHAPTER I

INTRODUCTION

Tactile sensing can help in inferring mechanical properties of objects. It can be used for creating a haptic map, which we define as *a set of pairs associating locations with haptic labels* [13]. However, because of the inherent local nature of tactile sensing, a naive approach of haptic mapping would require the robot to make physical contact with each and every location of interest which would be energetically expensive and time consuming. By using vision with tactile sensing, robots have the potential to haptically map their surroundings with greater efficiency. Our approach assumes that surfaces near a robot that are visually similar are more likely to have similar haptic properties. In our previous work [13], we introduced an iterative algorithm to infer dense haptic labels over a visible surface using sparse haptic labels by leveraging the complementary nature of the two sensing modalities. In [13], we showed that such an algorithm enables a robot to reach goal locations in a cluttered environment with fewer re-plans compared to when using tactile-only mapping method. In this work we introduce an improved algorithm (See Figure 1), to solve the same problem using a dense Conditional Random Field (CRF) [35]. The following section outlines the organization of the thesis.

In Chapter 2, we review the existing literature for work done in this field. In Chapter 3, we describe our previous algorithm [13] (Section 3.1 and our dense CRF based algorithm (Section 3.2) and compare the performance of the two in simulation (Section 3.4). Chapter 4, we describe the Convolutional Neural Network (CNN) used by Bell et al. [7] for material recognition (Section 4.1) and the fine-tuning procedure we used to adapt the network for our application (Section 4.2). We also compare

the difference in performance of the algorithm with and without the CNN to provide a prior probability distribution (Section 4.3). In Chapter 5, we describe the use of an entropy measure to intelligently sample the next point of contact and evaluate it. In Chapter 7, we discuss conclusions drawn from the work and list out possible future works. The algorithm introduced in this work, Chapters 2, 3 and 6, has been submitted as part of a paper to the IROS 2016 conference [50].

CHAPTER 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. Vision and touch are often seen as complementary modalities as vision gives information about the wide visible surface whereas touch can be used to infer properties in a small area of contact.

2.1 Material recognition using tactile sensing

Robots make contact with various objects in the environment while performing various manipulation tasks. Tactile sensing enables robots to gain information regarding various object characteristics such as surface texture [29, 38], stiffness [43] and temperature [14,39,42]. These properties have been shown to be useful in material classification [14,29,38]. Tactile sensing, due to its inherent local nature, can only provide 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.

2.2 Material recognition using vision

Vision has been used for texture recognition [37, 59]. Recent works have shown that vision can also be used for material recognition tasks [6, 7, 21, 28, 40]. Bell et al. [7] introduced a large scale database, Materials in Context Database (MINC), that has 23 material categories. They also introduced a framework that combines a fully convolutional neural network with a fully connected conditional random field (CRF)

to produce pixel level material labeling of the scene with 73.1% mean class accuracy. In this work we use the CNN model trained by Bell et al. [7] 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 [25, 48, 51]. Arnab et al. [6] used a joint dense CRF model to augment dense visual cues with spare auditory cues to estimate dense object and material labels. While a basic CRF uses a pairwise potential term that incorporate 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.

2.3 Integration of vision and touch

Studies have shown that under some conditions, humans can be modeled as combining visual and haptic information using a maximum-likelihood integrator [23]. They propose that humans integrate estimates of an environmental property through each individual sensory modality by performing an MLE integration. Some early work in integrating vision and haptics [4,53,62] integrated information from the two modalities to build models of objects.

Allen [4] 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 which was compared with a model database to recognize the object. Stanisfield [53] presented a robotic perceptual system which used vision to segment objects and haptically explore them to build a model of the object. Hosoda et al. [27] used a Hebbian network to learn consistency between data from a camera and tactile sensors to identify slip. Zytkow and Pachowicz [62] used vision and touch to learn

object manipulation tasks. Luo et al. [41] 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.

Ueda et al. [56] 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 [19], used a lightweight plunger and an optical mouse to take the surface image to classify the material. Zheng et al. [61] 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 Fully-Convolutional Neural Networks, one with images as inputs and the other with spectograms of acceleration signals as inputs and used a fully-connected layer to combine information from both. Gao et al. [26] also trained two CNN models for haptic and vision and combined the two using a fusion layer for classification of haptic adjectives. They used four exploratory behaviors such as *hold*, *squeeze*, *slow slide*, and *fast slide* to identify haptic signals from BioTac sensor. Our problem is different from this work, as we combine vision and tactile sensing to infer haptic properties of the entire visible surface.

2.4 Haptic Mapping of the scene

Haptic maps generated via active exploration [5, 24, 49] and incidental contact [8] tend to be sparse due to the local nature of tactile sensing. Our previous work [8, 13] generated haptic maps of the visible scene and demonstrated the usefulness of haptic map in manipulation tasks. 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 few points of contact. However, this algorithm did not leverage the potential of vision to predict

the haptic labels via visual cues alone. In this thesis, we introduce an improved algorithm to achieve the same goals.

CHAPTER III

GENERATING A DENSE HAPTIC MAP

As discussed in Chapter 1, this work considers the problem of inferring a dense haptic map of the visible surface by using vision, given sparse haptic labels assigned using tactile sensing. The basic assumption is that surfaces that are visually similar are more likely to have similar haptic properties. In Section 3.1, we discuss our previous approach [13] to solve the problem. In Section 3.2 we introduce a new approach to solve the problem using a dense CRF.

3.1 Previous Algorithm

Our previous algorithm [13] used the sparse haptic labels from tactile sensing 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 maintained a visual-haptic relation.

3.1.1 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.

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: $LCL\_list \leftarrow Location\ Color\ Label\ relation\ list$
while SH_Labels **is not empty do**
  $SH \leftarrow SH\_Labels.POP()$
  $XY \leftarrow SH.XY$
  $RGB \leftarrow RGB\_Im\,[XY]$
  $Label \leftarrow SH.Haptic\_Label$
  $LCL \leftarrow LCL\_list$ **entry with best match** XY
  if $LCL == None$ **then**
    $Label\_counts \leftarrow$ **new 0 array**
    $Label\_counts\,[Label] + +$
    $LCL \leftarrow (XY, RGB, Label)$
    $LCL\_list.Append(LCL)$
  else
    $LCL.Label\_counts\,[Label] + +$
  end if
  $CL \leftarrow CL\_list$ **entry with best match** RGB
  if $CL == None$ **then**
    $Label\_counts \leftarrow$ **new 0 array**
    $Label\_counts\,[Label] + +$
    $CL \leftarrow (RGB, Label\_counts)$
    $CL\_list.Append(CL)$
  else
    $CL.Label\_counts\,[Label] + +$
  end if
end while

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 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 3.1.2.

3.1.2 Dense Haptic Map Generation

In this stage, our algorithm (see Algorithm 2) uses CL_list (See Section 3.1.1) to infer the haptic labels of the rest of the visible scene. For this, the algorithm compares the

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: $LCL\_list \leftarrow Location\ Color\ Label\ relation\ list$
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**
        $Label \leftarrow$ "$Unclassified$"
    else
        $Label \leftarrow ArgMax\,(CL.Label\_counts)$
        $Count \leftarrow CL.Label\_counts\,[Label]$
        if $Count \leq 0.8 * Sum\,(CL.Label\_counts)$ **then**
            $Label \leftarrow$ "$Uncertain$"
        end if
    end if
    return $Label$
end procedure
for each $Pixel$ **in** RGB_Im **do**
    $RGB \leftarrow Pixel.RGB$
    $Pixel.Label \leftarrow GetLabel\,(RGB, CL\_list)$
end for
for each LCL **in** LCL_list **do**
    $RGB \leftarrow LCL.RGB$
    $Label \leftarrow GetLabel\,(RGB, CL\_list)$
    if $LCL.Label \neq Label$ **then**
        $XY \leftarrow LCL.XY$
        $C\_RGB \leftarrow ConnectedComponent(RGB\_Im, XY)$
        $C\_D \leftarrow ConnectedComponent(D\_Im, XY)$
        $Segment \leftarrow C\_RGB \cap C\_D$
        for each $Pixel$ **in** $Segment$ **do**
            $Pixel.Label \leftarrow LCL.Label$
        end for
    end if
end for
for each $Pixel$ **in** RGB_Im **do**
    $Hap\_map.Add\,(Pixel)$
end for

color of every point in the visible scene with the colors maintained in the CL_list. 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.

3.2 Using a dense CRF to generate Dense Haptic Map

We use a dense conditional random field (CRF) [35] in a manner similar to Arnab et al. [6] to obtain a dense haptic map. Given a dense probability map from the visual modality (p^v) (Described in Chapter 4) and a sparse probability map from tactile sensing (p^t), we combine the two probabilities using convex combination. This is inspired by Arnab et al. [6] who generated the material labels using two separate modalities, vision and audio. They combine these two terms by taking their convex combination. We use this approach as the tactile labels are sparse, similar to the labels acquired by audio sensing in [6]. We combine the two terms as shown in (1)

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

where $p_i^v(x_i)$ is the probability of the label generated by a classifier trained to

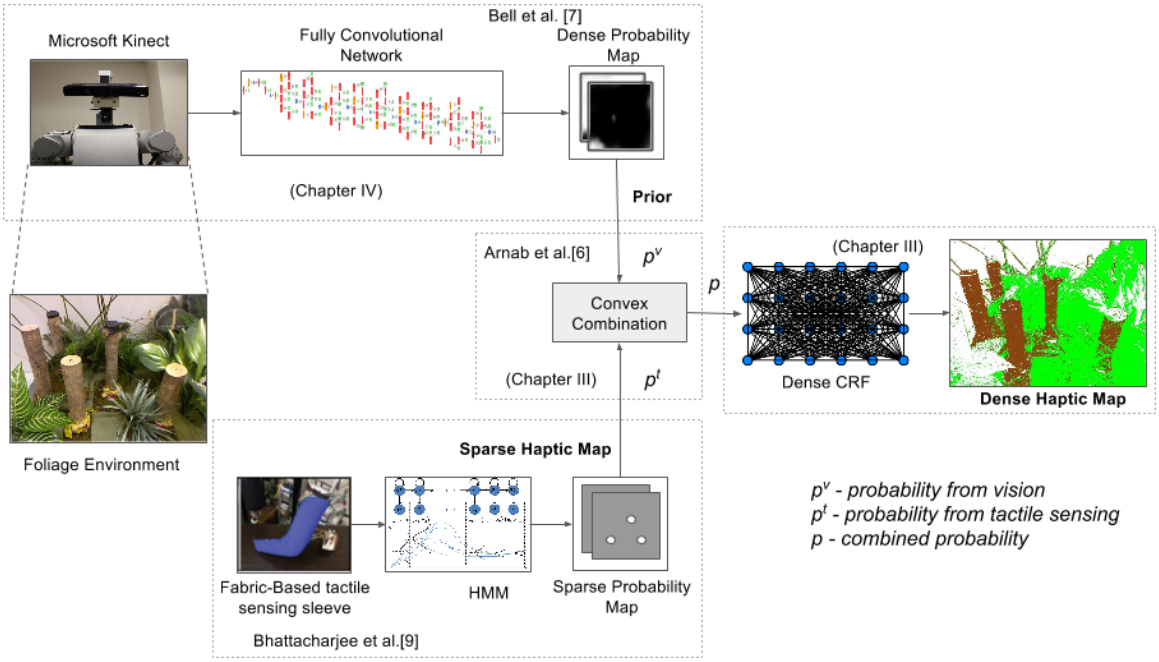

Figure 1: *Integrating tactile sensing and vision for material recognition. The vision pipeline is processed by a fully convolution network trained on the MINC dataset by Bell et al. [7]. We use a fabric based tactile sensing sleeve and HMMs [11] to classify points of contact based on their haptic property. We combine the probabilities from the two modalities by taking a convex sum inspired by Arnab et al. [6]. A dense CRF is used to predict labels for each pixel from the combined probability map.*

predict labels using vision (Section 4.1) and $p_i^t(x_i)$ is the probability of the label generated by the tactile sensing (True labels in Section 3.4 and labels generated by HMMs [9] in Section 6). U is a uniform distribution. w_{tv} and w_v are weight parameters (which can take a value between 0-1) which determine the importance of each individual prediction. We then assign haptic labels to individual pixels by using a dense CRF [35]. The Gibbs energy function for a dense CRF is as follows:

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

where $\psi_i(x_i)$ is the unary term and $\psi_{ij}(x_i, x_j)$ is the pairwise term. We use the unary and pairwise terms used by Bell et al. [7].

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

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

In (4), δ is a label compatibility term which introduces a penalty if two pixels are assigned different labels and k is a Gaussian kernel. The pairwise feature f_i used in [7] 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:

$$f_i = [\frac{p_i^x}{\theta_p d}, \frac{p_i^y}{\theta_p d}, \frac{I_i^L}{\theta_L}, \frac{I_i^a}{\theta_{ab}}, \frac{I_i^b}{\theta_{ab}}] \tag{5}$$

To summarize,

1. We use the the dense CRF framework used by Bell et al. [7]

2. We use the formulation used by Arnab et al. [6] to combine the probabilities from the two modalities.

3.3 Dataset for evaluation

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 [22, 31, 36, 44, 46]. We relabeled the segmented objects in these images using tools from [47], 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* and *Soap*. 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 [12, 15, 20, 30, 38, 55]. 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).

3.4 Comparison with our previous algorithm [13]

We compared the performance of this algorithm to our previous algorithm from [13] using the same evaluation procedure. We used the 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 described in Section 3.3. 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

Table 1: Comparison of performance of our current algorithm with our previous algorithm [13].

Contact Points/ No. of Objects	*Pixels correctly labeled*	
	Present Algorithm (*Avg.*±*StdDev*)%	*Previous Algorithm* (*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 2: Performance on different environments after 40 contact points per object using current algorithm.

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 %

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 $\sum_{i=1}^{186} 40 \times N_i = 52160$ labeled pixels in total. For each of the sampled points, we assumed that a patch (of size 10×10) 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 sampled pixel alone. We ran all our simulations with w_v and w_{tv} set to 0.001.

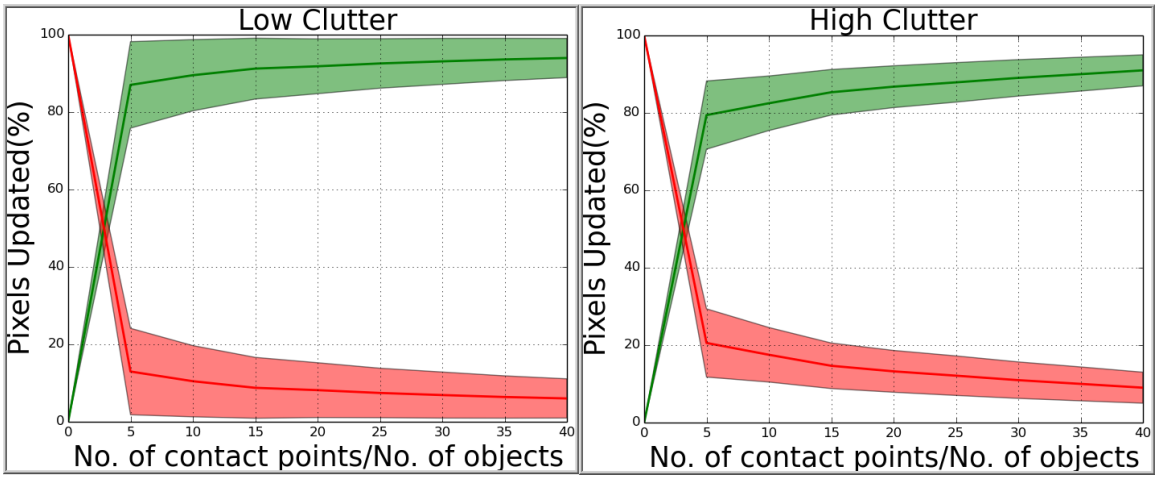

Figure 2: *Percentage of pixels assigned correct/incorrect labels for environments with different clutter densities. Green: Correct, Red: Incorrect.*

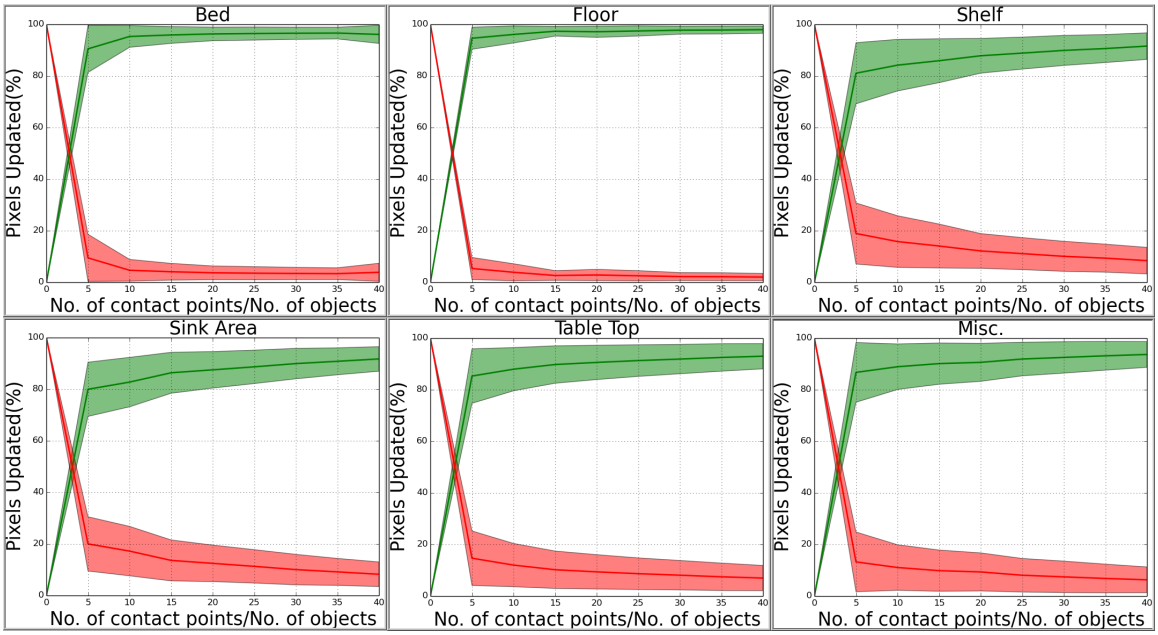

Figure 3: *Percentage of pixels assigned correct/incorrect labels for environments categorized based on scene. Green: Correct, Red: Incorrect.*

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. Table 1 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 pixels per object, the algorithm correctly updated an average of 81% of pixels which is higher than the results achieved for 40 pixels per object with our previous algorithm in [13].

3.4.1 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 2 and Figure 2 show the results. Our algorithm performed better with low-clutter environments (F_1 $score = 0.86$) when compared to high-clutter environments (F_1 $score = 0.72$).

3.4.2 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 3.4.1. Table 2 and Figure 3 show the results.

3.5 Implementation:

We implemented our algorithm in Python using scikit-image [58], NumPy [57] and OpenCV [17] libraries. We used the Python code provided by Bell et al. [7], which

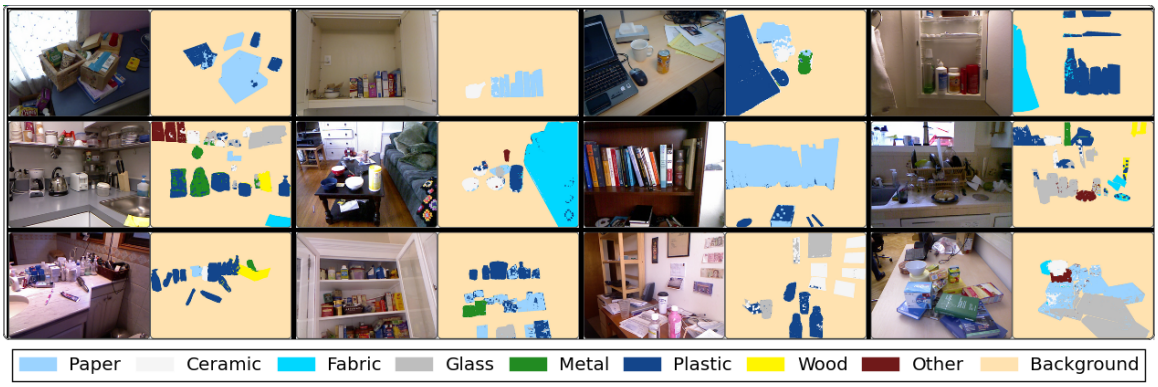

Figure 4: *Simulation results of haptic categorization for some example images from different publicly available datasets [22, 31, 36, 44, 46]. These examples show scenes from different environments with varying density of clutter.*

uses Caffe [32] for building CNN and the C++ implementation of dense CRF released by Krähenbühl et al. [35].

CHAPTER IV

MATERIAL RECOGNITION USING VISION

In Chapter 3 we described how we uses a dense CRF (Section 3.2) to generate dense haptic labels given the material probabilities generated using tactile sensing and vision. In this chapter we discuss how vision is used to generate the material probabilities. In Section 4.1, we describe the CNN model, released by Bell et al. [7], trained for material recognition. In Section 4.2, we describe the procedure adopted to fine tune the model for our application.

4.1 Material Recognition using CNN

Bell et al. [7] introduced the Materials in Context Database (MINC), a large-scale open dataset of material in the wild. The images in the MINC dataset are annotated with 23 material categories (*brick, carpet, ceramic, fabric, foliage, food, glass, hair, leather, metal, mirror, painted, paper, plastic, polishedstone, skin, sky, stone, tile, wallpaper, water, wood, other*). Bell et al. [7] also released the fine tuned GoogLeNet model [54] on patches extracted from the MINC dataset. Figure 5 shows the architecture of the CNN model.

4.2 Fine tuning the CNN model

For this work we fine tuned this network to recognize 8 material categories (*Ceramic, Paper, Plastic, Metal, Fabric, Wood, Glass and Others*) using patches extracted from various RGB-D datasetes [6, 31, 34, 36, 52, 60]. We believe that these labels could reasonably be classified using tactile sensing and they better match the labels in the MINC database [7]. We annotated 228 images from these publicly available datasets with 8 material categories mentioned. We ensured that none of these 228 images were

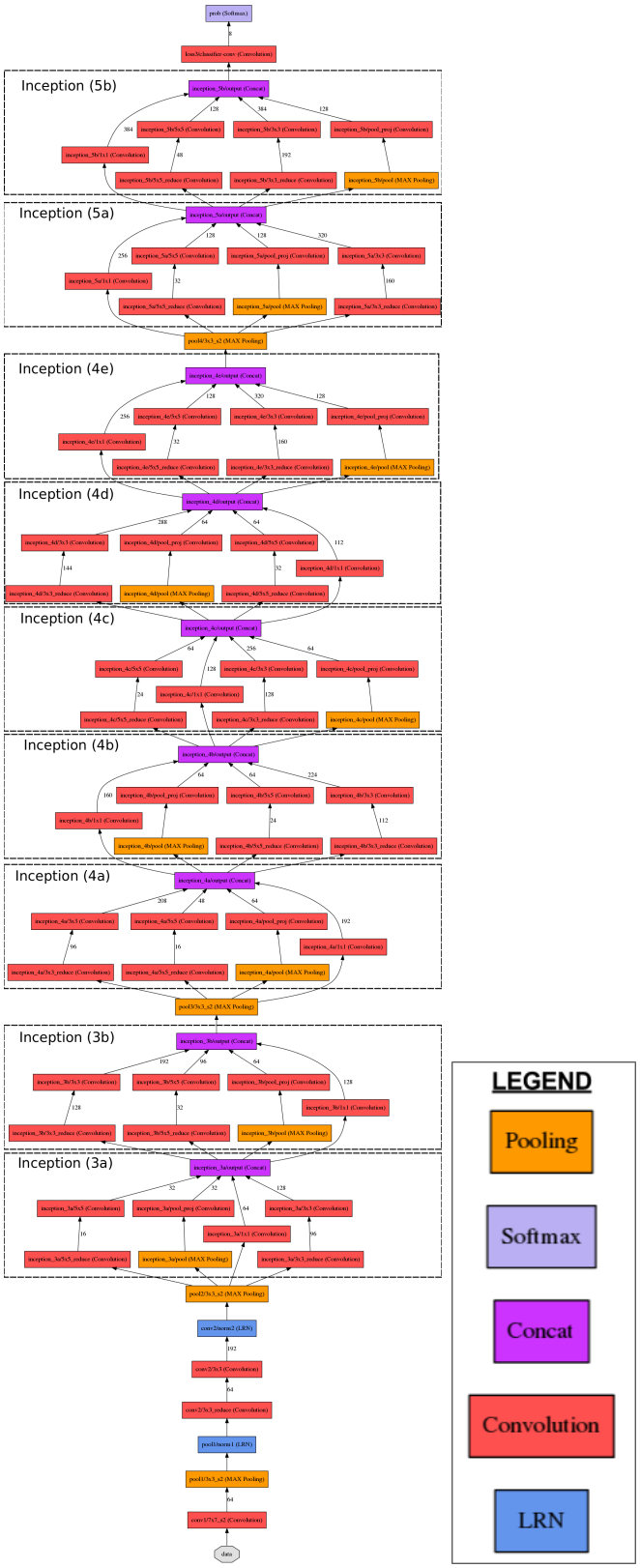

Figure 5: *The fully convolution network fine tuned on our image dataset. This GoogLeNet [54] model was originally fine tuned on the MINC dataset by Bell et al. [7].*

Table 3: Effect of prior on performance using current algorithm.

Contact Points	*Pixels correctly labeled*	
	Uniform Prior (*Avg.*±*StdDev*)%	*Prior 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 %**

part of the 186 image dataset. We then adopted the same procedure used by Bell et al. [7] 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 dimensions of the patch was 23.1 % of the smaller image dimension. We then fine tuned the MINC CNN model using caffe [32]. We replaced the last fully connected layer 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 (See Figure 5). We used one tenth of the learning rate used for the last layer for the rest of the convolution layers. We used the stochastic gradient descent (sgd) method for optimization. The training parameters are presented in Appendix A.

4.3 Effect of prior probability p^v

For this set of simulations, we made two changes to the procedure adopted in Section 3.4. First we redefined the haptic labels of the 186 images as *Ceramic, Paper, Plastic, Metal, Fabric, Wood, Glass and Others*. We believe that these labels could reasonably

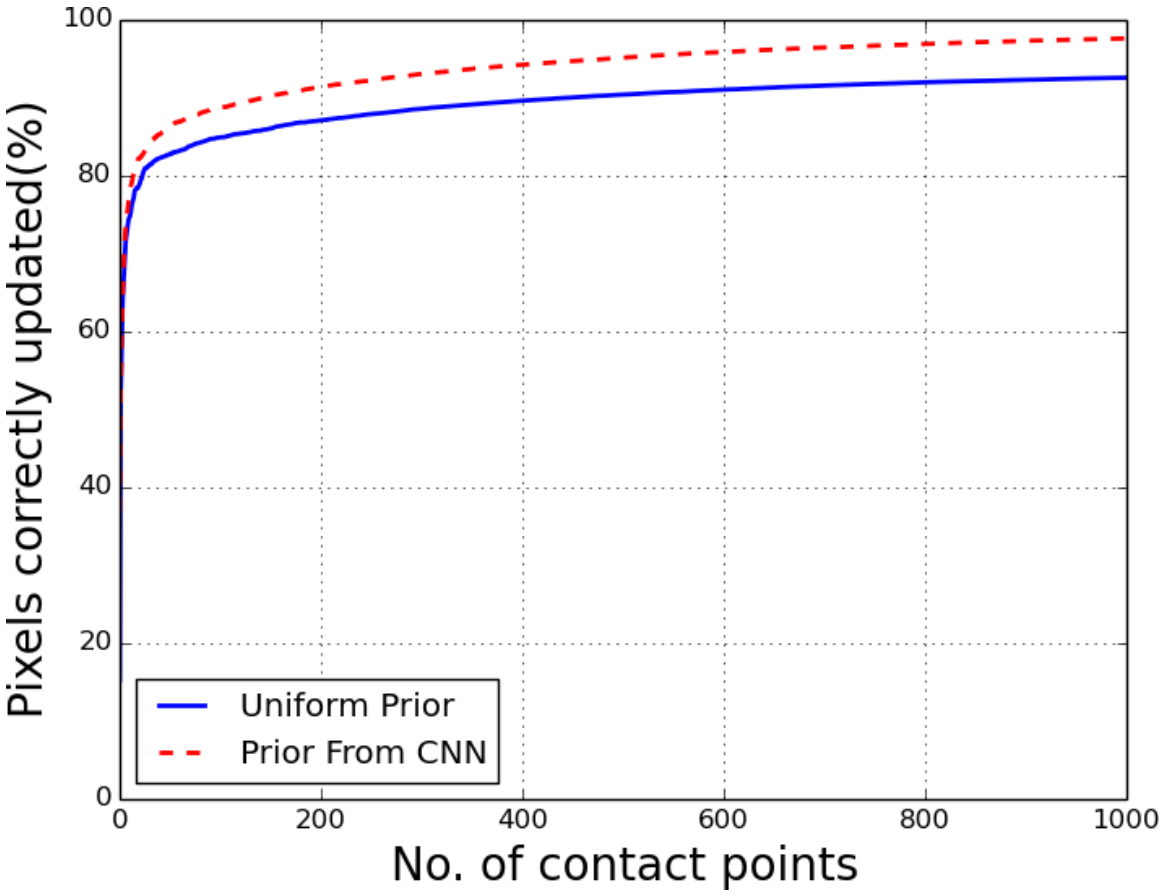

Figure 6: *Effect of prior on performance of the algorithm.*

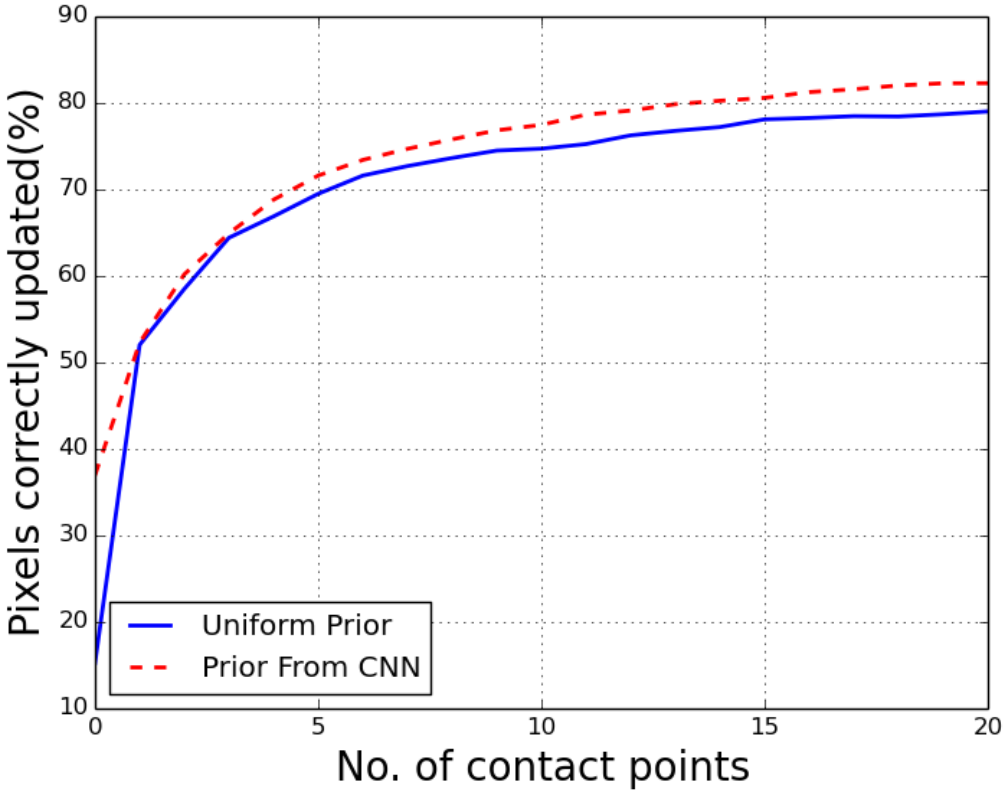

Figure 7: *Figure 6 zoomed in for first 20 contact points.*

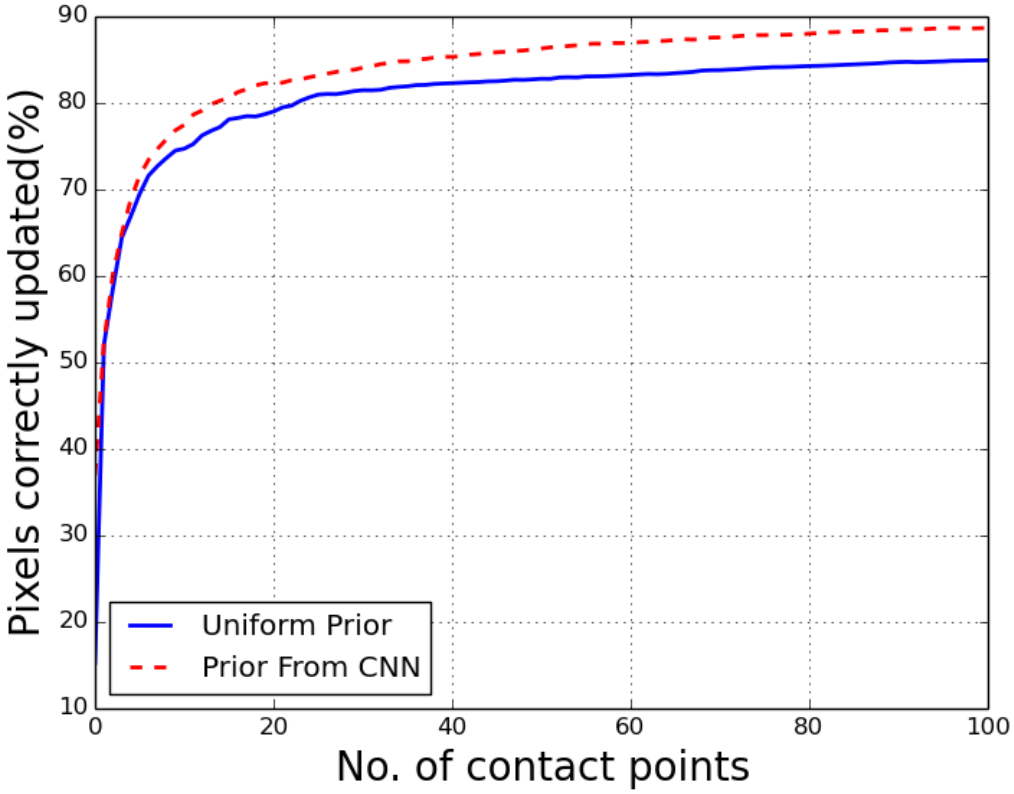

Figure 8: *Figure 6 zoomed in for first 100 contact points.*

be classified using tactile sensing and they better match the labels in the MINC database [7]. Second, we sample the points of contact directly from the image instead of sampling from the pool as done in Section 3.4.

We evaluated the effect of the prior (p_v) by comparing the performance of Uniform prior with the the prior generated using the fully convolutional network released by Bell et al. [7]. We repeated the same simulation as described in 3.4 using the same set of 186 images. We performed the simulation by initializing p^v to two different values described below:

4.3.1 Uniform Distribution

We set p^v to uniform distribution, i.e., we assume that the robot has no knowledge of the class of the pixels. We assign equal probability to all classes.

4.3.2 Prior determined by CNN

We set p^v to the probability map of the image using the fine tuned CNN described in Section 4.2.

The results for the two different setting are shown in table 3 and in Figures 6, 7 and 8. The dense CRF using uniform prior assigns the correct labels to 15.27% of pixels. On the other hand the dense CRF using the prior from CNN assigns the correct labels to 36.96% of pixels. We see considerable improvement in the performance of the algorithm when it uses the prior generated from the CNN.

CHAPTER V

EXPLORATION USING ENTROPY MEASURE

In this work, we assume that robot is actively seeking for tactile information by making contact with the environment. In chapter 3 and 4, the simulations were conducted by randomly sampling the next point of contact. This strategy of sampling the points of contact may not be the most optimal in terms of gaining the most amount of information after each point of contact. Besides random sampling, there are various entropy based strategies for exploration [16,18,45]. In these approaches, the idea is to actively take actions that reduces entropy, a measure of uncertainty. In this chapter we evaluate the use of entropy measure on the probability map defined in Eq. 1, to sample the next point of contact. In Section 5.1, we describe the procedure used to generate the entropy measure and Section 5.2, evaluates this procedure in simulation.

5.1 Entropy Measure

Given a probability map p_i as defined in eq. (1), we calculate the entropy as shown in Eq. 6,

$$e_i = \sum_{x_i} p_i \times log(p_i(x_i)) \tag{6}$$

We use a mean spatial filter to filter the entropy. We then choose the point with maximum entropy i.e., highest uncertainty as our next point of contact.

5.2 Evaluation

We repeat the simulation as described in Section 4.3 the only change being that the we sample based on entropy measure described above. The results are reported in Tables 4 and 5 and Figures 9 and 10. In case of Uniform prior (Figure 10), we observe

Table 4: Effect of sampling method with prior from CNN.

Contact Points	*Pixels correctly labeled*	
	Random Sampling (*Avg.*±*StdDev*)%	*Max. Entropy Sampling* (*Avg.*±*StdDev*)%
0	36.96 ±28.28 %	36.96 ±28.28 %
100	88.78 ±9.11 %	78.28 ±17.69 %
200	91.37 ±7.18 %	86.33 ±13.88 %
300	93.05 ±5.9 %	91.16 ±10.72 %
400	94.25 ±5.06 %	94.55 ±7.18 %
500	95.2 ±4.3 %	96.54 ±4.89 %
600	95.87 ±3.78 %	97.72 ±3.42 %
700	96.47 ±3.28 %	98.38 ±2.5 %
800	96.96 ±2.87 %	98.85 ±1.81 %
900	97.32 ±2.57 %	99.11 ±1.38 %
1000	**97.65 ±2.28 %**	**99.31 ±0.97 %**

that the max entropy sampling strategy improves the performance of the algorithm substantially. In case of prior being generated from CNN (Figure 9), the performance drops in case of max entropy based sampling. This could be because the CNN prior may not represent the uncertainty in the prediction accurately.

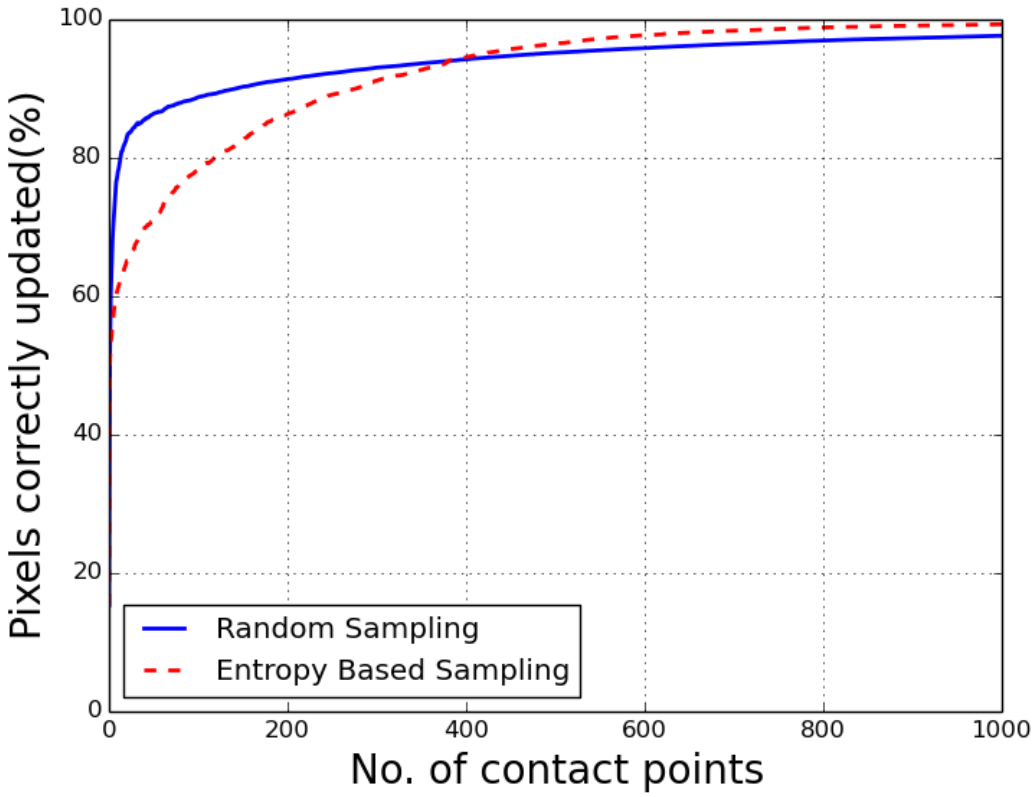

Figure 9: *Effect of sampling method on performance of the algorithm with prior from CNN.*

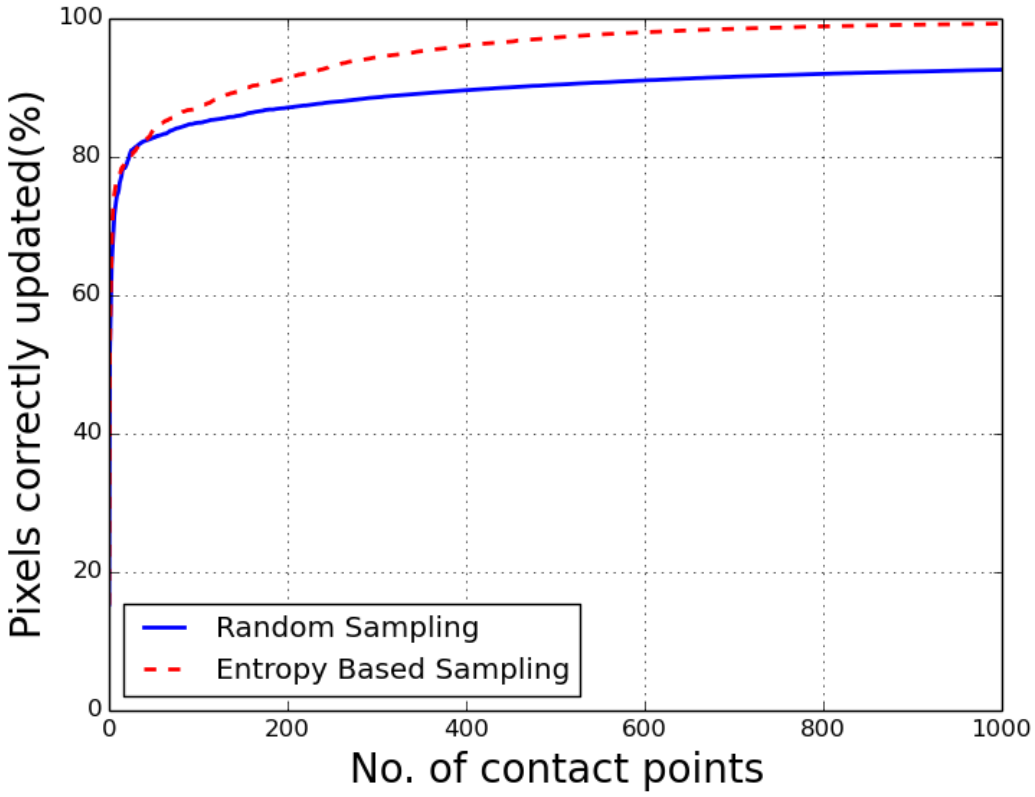

Figure 10: *Effect of sampling method on performance of the algorithm with uniform prior.*

Table 5: Effect of sampling method with Uniform prior.

Contact Points	*Pixels correctly labeled*	
	Random Sampling (*Avg.*±*StdDev*)%	*Max. Entropy Sampling* (*Avg.*±*StdDev*)%
0	15.27 ±25.66 %	15.27 ±25.66 %
100	84.94 ±16.08 %	87.11 ±11.18 %
200	87.12 ±15.47 %	91.42 ±8.05 %
300	88.6 ±15.11 %	94.4 ±5.67 %
400	89.63 ±14.93 %	96.1 ±4.35 %
500	90.43 ±14.83 %	97.25 ±3.34 %
600	91.05 ±14.79 %	98.02 ±2.47 %
700	91.59 ±14.76 %	98.51 ±1.92 %
800	92.0 ±14.75 %	98.86 ±1.42 %
900	92.32 ±14.74 %	99.1 ±1.12 %
1000	**92.6 ±14.73 %**	**99.25 ±0.92 %**

CHAPTER VI

EVALUATION WITH A REAL ROBOT

We performed experiments using a real robot to validate the algorithm and evaluate the performance while performing manipulation in a cluttered environment.

6.1 Experimental Setup

We used the humanoid robot DARCI (Figure 11), 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-sleeve [10]. The tactile sleeve has 25 discrete taxels that records the contact force. We trained HMM's for haptic category predictions [11]. 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. The robot had a Microsoft Kinect mounted on top of its torso. For our experiments, we used a system that runs Ubuntu 12.04 32-bit OS with the 3.5.0-54-generic Linux kernel. It has 16 GB RAM and an Intel® Core™ i7-3770 CPU @ 3.40 GHz × 8 processor. We used ROS Fuerte [1] for communicating with the RTPC and the robot DARCI. We used cv bridge [2] to convert between ROS images and OpenCV images. We used the GHMM toolkit [3] to implement and train the HMMs.

We tested the algorithm in the foliage environment, which is an artificially created cluttered environment. The environment is composed of trunks and leaves as seen in Figure 11. This setup was used in our previous work [13].

Figure 11: *A robot DARCI, equipped with a tactile-sensing sleeve (blue) and a Kinect, reaching into the cluttered foliage environment.*

6.2 Experimental Procedure

We programmed the robot DARCI to make 10 reaches (5 goal positions $\times$ 2 times) into the foliage environment for each trial. We conducted three such trials with different leaves in the foliage (See Figure 12). For each of the trials, we randomized the order in which the goal positions were selected. After reaching each goal position, the robot arm came back to the initial starting position and then moved to the next randomly selected goal position. The initial starting position of the robot arm was the same for all trials. The base of the robot was fixed during the entire experiment.

During each reach, the robot uses our previously developed dynamic MPC controller [33] to quickly reach the goal location with low contact forces. In this process the robot makes incidental contact with various points in the environment. We used forward kinematics to locate the contact points and transformed the co-ordinates of the points of contact to the image pixel co-ordinates using the camera properties and depth information. We ignored contacts beyond visible surface. We identified those

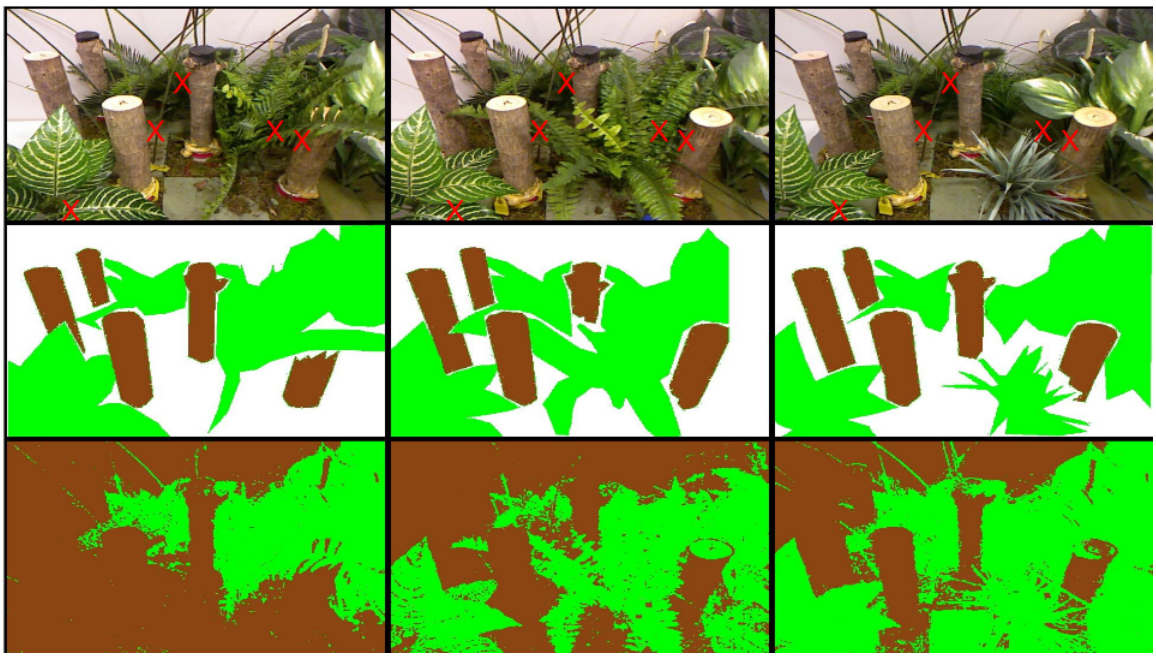

Figure 12: *The top row shows the scene after the robot made 10 reaches. The middle row shows the annotated images. The bottom row shows the corresponding haptic map. The trunks are marked with brown, the leaves are marked with green and the background is marked in white. We ignored the background for our evaluation. The Goal locations are marked with red X's in the top row.*

contacts as valid contacts, for which the depth of the contact point using forward kinematics is larger than depth of the visible surface from the depth image. We used trained left-right HMMs with 10 states and 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 the haptic property of the rest of the scene using our algorithm.

6.3 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 12 shows the haptic maps generated after various trials.

Table 6: Performance of the algorithm on the foliage environment using current algorithm.

Number of reaches (N)	Average number of contact points	*Pixels correctly labeled* ($Avg.\pm StdDev$)%
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 %

Ignoring the background, we evaluated what percentage of the pixels that belong to trunks and leaves are assigned the correct labels after each reach. The results are reported in Table 6. The algorithm 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 tactile recognition system during evaluations with a real robot. Also in some of the reaches, though the robot made contact with objects in the environment, the HMM failed to classify the haptic label of the object.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

In this chapter we summarize the contribution of this work and conclusions that can be drawn from this work. This chapter also proposes the future work that can be undertaken.

7.1 Conclusion

We presented a dense CRF based method to obtain dense haptic maps across visible surfaces using sparse haptic labels provided by tactile sensing. This method performed substantially better than our previous algorithm [13]. 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 [13]. We discussed the effect of 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 93% of the pixels in the images. The algorithm can also reach an average F_1 *score* of 0.86 for low-cluttered environments and 0.92 for bed 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 showed that the algorithm performs better with a prior probability generated from a CNN trained for material recognition. This framework also enables a robot to employ entropy based strategies to explore its environment. We also evaluated the algorithm on a real robot in the foliage environment where it assigned the correct label to 82.52% of

pixels after 10 reaches.

7.2 Future work

The work done as part of this thesis has shown significant improvement in performance over our previous algorithm for dense haptic mapping. However there is scope for improvement in various aspects of the algorithm. The following subsections outlines the possible future directions to the work.

7.2.1 Dynamic Environment

Our current algorithm assumes that the environment is mostly static and hence assigns the information acquired through tactile sensing to a physical location in the environment rather than the object. It will be a value addition to associate the point of contact to the object and track the object in order to enable the use of the algorithm in a dynamic environment.

7.2.2 Fine tuning the CNN

In this work, we fine tuned the CNN provided by Bell et al. [7], using a small dataset. It will be useful to create a larger dataset with material annotated to help the network generalize better in robotic application.

7.2.3 Different features for visual similarity

Besides the use of color, it should be possible to use other features such as texture to determine visual similarity between different pixels. This might require the use of higher resolution cameras which can help extract better texture features.

7.2.4 Other modality for material recognition

Besides the use of a standard RGB sensor to determine the prior probability, it would be possible to use other sensors such as a thermal sensing camera which can measure

the thermal emissivity to determine material properties of the environment. This information can then be added to our existing prior probability.

APPENDIX A

CAFFE SETUP FOR FINE TUNING OF CNN

solver.prototxt:

```
net: "finetuning/train_val.prototxt"
test_iter: 100
test_interval: 1000
# lr for fine-tuning should be lower than when starting from scratch
base_lr: 0.001
lr_policy: "step"
gamma: 0.1
# stepsize should also be lower, as we're closer to being done
stepsize: 20000
display: 20
max_iter: 100000
momentum: 0.9
weight_decay: 0.0005
snapshot: 1000
snapshot_prefix: "finetuning/tactile_vision"
# uncomment the following to default to CPU mode solving
# solver_mode: CPU
```

REFERENCES

[1] "ROS Fuerte." `http://wiki.ros.org/fuerte`.

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

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

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

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

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

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

[8] Bhattacharjee, T., Grice, P. M., Kapusta, A., Killpack, M. D., Park, D., and Kemp, C. C., "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.

[9] Bhattacharjee, T., Kapusta, A., Rehg, J. M., and Kemp, C. C., "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.

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

[11] Bhattacharjee, T., Kapusta, A., Rehg, J. M., and Kemp, C. C., "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*, pp. 219–226, IEEE, 2013.

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

[13] Bhattacharjee, T., Shenoi, A. A., Park, D., Rehg, J. M., and Kemp, C. C., "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.

[14] Bhattacharjee, T., Wade, J., and Kemp, C. C., "Material recognition from heat transfer given varying initial conditions and short-duration contact,"

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

[16] Bourgaul, F., Makarenko, A. A., Williams, S. B., Grocholsky, B., and Durrant-Whyte, H. F., "Information based adaptive robotic exploration," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, pp. 540–545, IEEE, 2002.

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

[18] Burgard, W., Fox, D., and Thrun, S., "Active mobile robot localization by entropy minimization," in *Advanced Mobile Robots, 1997. Proceedings., Second EUROMICRO workshop on*, pp. 155–162, IEEE, 1997.

[19] Charniya, N. N. A. and Dudul, S. V., "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.

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

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

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

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

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

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

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

[27] Hosoda, K., Tada, Y., and Asada, M., "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, pp. 111–115, IEEE, 2002.

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

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

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

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

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

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

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

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

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

[37] Leung, T. and Malik, J., "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.

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

[39] Lin, C. H., Erickson, T. W., Fishel, J., Wettels, N., Loeb, G. E., and others, "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*, pp. 129–134, IEEE, 2009.

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

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

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

[43] Murayama, Y., Constantinou, C. E., and Omata, S., "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.

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

[45] Otte, S., Kulick, J., Toussaint, M., and Brock, O., "Entropy-based strategies for physical exploration of the environment's degrees of freedom," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 615–622, IEEE, 2014.

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

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

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

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

[50] Shenoi, A. A., Bhattacharjee, T., and Kemp, C. C., "A crf that combines tactile sensing and vision for haptic mapping," in *Submitted to Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[51] Shotton, J., Winn, J., Rother, C., and Criminisi, A., "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.

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

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

[54] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

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

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

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

[58] van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014.

[59] Varma, M. and Zisserman, A., "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.

[60] Zhang, Q., Song, X., Shao, X., Shibasaki, R., and Zhao, H., "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*, pp. 193–200, IEEE, 2013.

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

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