Enhancing Haptic Distinguishability of Surface Materials with Boosting Technique

Priyadarshini K[1,2] and Subhasis Chaudhuri[1]

Abstract— Discriminative features are crucial for several learning applications, such as object detection and classification. Neural networks are extensively used for extracting discriminative features of images and speech signals. However, the lack of large datasets in the haptics domain often limits the applicability of such techniques. This paper presents a general framework for the analysis of the discriminative properties of haptic signals. We demonstrate the effectiveness of spectral features and a boosted embedding technique in enhancing the distinguishability of haptic signals. Experiments indicate our framework needs less training data, generalizes well for different predictors, and outperforms the related state-of-the-art.

I. Introduction

Different sensory inputs (vision, smell, touch, etc.) are crucial to endow robots with human-like understanding of the physical world. For instance, when we hold an object, we see its color and shape, feel its texture, and smell its scent; our brain binds these perceptions together to form the concept of that object. While there have been extensive studies in the vision domain, the learning problem is relatively unexplored in the haptics domain.

For many robotics applications, the haptic sensation is crucial for manipulating objects in the surrounding. As a result, a great body of research in recent years [22], [16], [8], [20], [18] have focused on understanding tactile properties such as surface friction, compliance (soft vs. hard), and the fragility of objects. Besides haptic adjectives, surface material classification is also relevant for robot identifying, grasping, and characterizing objects. Existing methods of classifying tactile properties of objects either use input signals recorded from very high-end sensors [3] or rely on multimodal features for surface material classification [22], [16], [8], [20]. While Sinapov *et al.* [18] use single modality acceleration data to generate discriminative features for material classification, they achieve very low classification accuracy ($\sim$30-40%). The accuracy improves only with the inclusion of other modalities such as image, sound, friction, and scanning velocity [22], [8]. However, this requires additional sensing hardware for data collection, and that reduces the generalizability of algorithms. Moreover, the feature vectors in some of these works are also quite high-dimensional (e.g., 125-D in [18]), which can lead to overfitting. We present a metric-based feature transformation technique that learns a set of discriminative tactile features by projecting *single* modality acceleration data of surface materials to an embedding space where the distinguishability between different classes of haptic signals is well realized.

Most often, in the raw input space, no underlying pattern is easily visible within the data. The data is typically too high-dimensional, and different classes of signals are not well-separated in this space. Neural network models are commonly used to capture the complex underlying relationship in the data. However, the success of such methods can be largely attributed to the powerful computing systems and abundance of training data. The latter is often a limiting factor in the haptics domain. Moreover, highly parameterized techniques such as neural networks lack generality, adaptability, and robustness in the case of smaller datasets.

We propose a **general computational framework**, founded on **spectral features** for haptic textures generated only from acceleration data, augmented by a **boosted embedding** technique trained with discrete label-based supervision. Such embedding has several important benefits. The performance of various pattern recognition algorithms, such as classification and clustering, can improve significantly after re-scaling the data using a learned embedding [24]. Depending upon how the embedding function is learned, the projected data in the learned space reflects the distinguishability between signals. Our learning framework integrates two crucial components: (1) discriminative feature extractor, (2) metric-based feature transformer. We apply the spectral transformation to the input signals and extract representative components using the binning operation to learn discriminative compact features from high-dimensional acceleration signals. We call this constant Q-factor filter bank (CQFB) feature, and it's briefly described in [14]. Next, we improve the distinguishability of haptic signals by transforming the extracted features using a linear projector called BoostMetric[17]. Our experiments demonstrate that the CQFB feature augmented with the boosted embedding gives much better performance of semantic discrimination tasks such as classification and clustering than without embedding.

II. Related Work

Distinguishing between surface materials using haptic touch is difficult due to the presence of high-frequency vibrations component in the texture signals. Several prior works use different discriminative textural features such as roughness, friction, and hardness of materials for classification tasks [8], [20], [22], [21], [4], [18], [15].

[1,2]Priyadarshini K is with Department of Electrical Engineering, IIT Bombay and Sony AI, Tokyo priyadarshini.kumari@sony.com

[1]Subhasis Chaudhuri is with Department of Electrical Engineering, IIT Bombay sc@ee.iitb.ac.in

Strese *et al.* [20], [22], [8] use multi-modal features (tactile and visual) to improve the performance of the classifier. Authors in [25], [8] employ the powerful capability of the deep neural network to learn automated discriminative features, again using multi-modal sensory inputs. Joseph *et al.* [16] and Strese *et al.* [22] report very low classification accuracy ($\sim$30-40%) when haptic data (acceleration signal) alone is used for feature extraction. Joseph *et al.* [16] and Jeremy *et al.* [7] show that successful recognition of surface material depends on scanning velocity and normal force applied during recording. Strese *et al.* [21] mitigates this effect by proposing a deep neural network approach that learns features from acceleration and tapping data. Although the input is just a haptic modality, acceleration trace and tapping data require a separate recording procedure.

Another exciting line of work [18], [7], [12] shows along with tactile information, the prior knowledge of exploratory movements is also needed to characterize the objects. A study by Lederman *et al.* [12] shows significant improvement in KNN classification accuracy (from 48% to 68%) using multiple exploratory movements. Further, Jeremy *et al.* [7] build an advanced model that adaptively selects the optimal feature and the optimal exploratory movements. Moreover, they use a high-end BioTac sensor for haptic data recording, which has similar mechanical properties as a human fingertip. They report a classification accuracy of around 95% on 117 textures. In contrast to our approach, all methods use either multi-modal or high-end data recording sensors such as a biomimetic sensor and multiple exploratory movements for feature generation. Our approach tries to classify 69 surface materials using the **acceleration data alone**, which is recorded using a low-cost acceleration sensor (LIS344ALH) in a **single exploration**. The dimension of our feature vector is also low (only 10-13). Moreover, our method provides a formal and general representation of haptic textures that directly captures the relationship between different signals in terms of similarity or dissimilarity.

The closest work to ours is [6], where embedding space is learned using MDS-based projection approaches. While these technique has been widely used for several applications including clustering and modeling perceptual dissimilarity between objects, they are non-parametric and cannot compute a suitable projection of a new signal with new features at test time.

III. METHOD

In this section, we present the various stages of our framework. Our objective is to learn the metric-based feature transformation function over the input space to enhance the distinguishability of haptic textures. Our data consists of traces of acceleration signals of different surface materials such as stone, wood, grass, paper, etc. The learned transformation function re-scales the data in the embedding space such that the dissimilar haptic signals (say stone and grass) are placed far apart, and a similar pair (say stone and granite) are placed close together.

To reduce the dimension of data, we combine the three components of the acceleration signal $y(t)$ into a single vector representing the spectral magnitude $Y(w)$ using DFT321 [11]. As mentioned in [11], the method preserves the spectral characteristic of the acceleration signal even after reducing the dimension. The psychophysical studies show that touch is highly sensitive up to 1 kHz frequency, and the optimal sensing frequency is around 250 Hz [19]. We thus have taken a signal spectrum $Y(w)$ within the frequency of 1 kHz for feature extraction. Each component of our algorithm is explained in the following sections.

Feature extraction: The major problem in learning discriminative features with time series data is the phase invariant compact representation. Moreover, the distance metric used for feature transformation heavily relies on the meaningful feature to place similar data close to each other and dissimilar data far apart. However, in a high dimensional space, like in the case of time-series data, the contrast between similar and dissimilar data decreases [2]. Therefore it is essential to select the right feature set which can compactly represent the input data, keeping only important information while discarding the noise and redundant data. The frequency-domain analysis of texture signals is useful in terms of the reduction of correlated data or noise. Hence in our learning algorithm, we use spectral features.
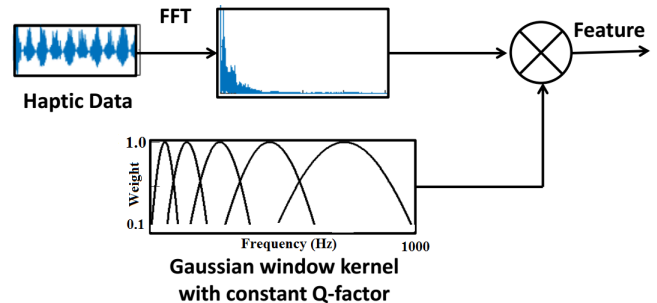

Fig. 1: A variable bandwidth Gaussian shaped filter bank with constant Q-factor is used for feature extraction.

We extract the features from the signal spectrum $Y(w)$ by dividing them into N bins of varying sizes. Each component of the feature vector indicates the total spectral energy within each bin. Since the human sensitivity to vibration falls exponentially as frequency increases [1], we propose to vary the bin size in geometric progression as one moves from low frequency to high frequency (as shown in Fig. 1). Thus, low-frequency components get more prominence than high-frequency components and reduce the effect of noise in the data. Further, instead of using a rectangular window to integrate the spectral components within a bin, we use a window with an appropriate width of σ to provide a certain spectral overlap between adjacent bins.

We apply a variable bandwidth Gaussian shaped filter bank $\{W_j(w)\}_{j=1}^N$ on frequency spectrum of the signal $Y(w)$ as shown in Fig. 1. The sigma(σ) and center frequency (f_c) of Gaussian kernel is varied in geometric progression with a constant Q factor α. At α equal to one, we get a

regularly spaced, constant bandwidth spectral bin. In order to keep the Q-factor constant, we vary the center frequency (f_c) as well as σ of the Gaussian kernel in geometric progression as follows-

$$f_{c_j} = \alpha^{j-1} f_{c_1}; \quad \sigma_j = \alpha^{j-1}\sigma_1; \quad j \in [1,N] \qquad (1)$$

where N denotes the dimension of the feature vector. Since the central frequency of the last Gaussian window(f_{c_N}) cannot exceed the maximum frequency of spectrum (f_{max}), we have fixed f_{c_N} equal to f_{max}. Thereafter we compute other required parameters of the Gaussian filter bank as follow-

1) First, we find the center frequency of the first Gaussian window (f_{c_1}) using Eq. 1, *ie*, $f_{max} = \alpha^{N-1} f_{c_1}$.
2) Now we compute the sigma of the first Gaussian window (σ_1) such that its passband dies down to 10% level at zero frequency, *ie*, $e^{-\frac{f_{c_1}^2}{2\sigma_1^2}} = 0.1$.

Having estimated f_{c_1} and σ_1, one can find f_{c_j} and $\sigma_j \quad \forall \quad j$ using Eq.1. One may note that $\frac{f_{c_j}}{\sigma_j}$ is the same for all j and hence the Q factor is constant. The computed j^{th} component of feature is given by

$$a_j = \int_{f_{j-1}}^{f_j} W_j(w)|Y(w)|^2 dw \quad j \in [1,N] \qquad (2)$$

We call the derived feature $x = \{a_1, a_2, ... a_N\}$ as constant Q-factor bandwidth (CQFB) feature. The next step of our algorithm is to learn an appropriate metric-based feature transformation for better distinguishability of haptic textures.

Feature transformation using boosting technique: Our goal is to learn a feature transformation matrix $M \in \mathbb{R}^{\mathbb{N} \times \mathbb{N}}$ such that in the projected space, the Euclidean distance between two haptic signals indicates the dissimilarity between them. We use the boosting-like technique proposed by Chunhua *et al.* [17] for linear metric-based feature transformation. The idea is based on the interesting insight that any positive semi-definite matrix such as Mahalanobis distance matrix can be decomposed into a linear positive combination of trace-one-rank-one matrices. Therefore an efficient metric can be learned efficiently by combining multiple weak learners well-suited for small datasets.

Estimating the Mahalanobis distance $d_M(x_i, x_j)$ is equivalent to first transforming the data into a new space using a linear projector $x \to L^T x$ and then applying the Euclidean metric to the transformed data. One of the major challenges in learning the Mahalanobis matrix is to ensure the positive semi-definiteness constraint on the learned matrix M. To ensure positive semi-definiteness, sometimes transformation matrix $\mathbf{L}$ is learned instead of $\mathbf{M} = LL^T \in \mathbb{R}^{N \times N}$ [9], [23]. But this approach does not ensure the global optimum value as the cost function becomes non-convex in this case. The standard SDP (semidefinite programming) solver can also be used to ensure the global optimum, but SDP is inefficient for metric learning, especially when the feature dimension is high. The boosting technique [17] addresses this challenge by incorporating the properties of a positive semi-definite matrix in its learning approach itself.

The method combines the trace-one and rank-one matrix iteratively, and the final metric $\mathbf{M} \in \mathbb{R}^{N \times N}$ takes the form of $\mathbf{M} = \sum_{k=1}^{K} w_k \mathbf{U_k}$. The w_k is the non-negative weight and the matrix $\mathbf{U_k}$ is a trace-one, rank-one matrix. In this way, the method incorporates the positive-definiteness constraint elegantly without increasing the computational complexity.

The input to the boosting algorithm requires a set C of order relations among data.

$$\mathbf{C} = \{(x_i, x_j, x_k) \mid d_M(x_i, x_j) < d_M(x_i, x_k)\} \qquad (3)$$

Here the training samples x_i and x_j belong to the same class, and x_k belongs to a different class. The set of constraints $\mathbf{C}$ is in the form of triplets of training samples where the sample x_i is closer to sample x_j than x_k. In order to create well separated compact clusters of different classes, we maximize the distance margin $\rho_r = (x_i - x_k)^T \mathbf{M}(x_i - x_k) - (x_i - x_j)^T \mathbf{M}(x_i - x_j)$ between the pair (i,j) and (i,k) of each triplet r in the Mahalanobis distance metric space.

As mentioned earlier, the aim is to maximize the distance margin (ρ_r) with an appropriate regularization. The optimization problem is defined [17] using an exponential loss function as follow

$$\begin{aligned} \underset{\mathbf{M}}{\text{minimize}} \quad & \log(\sum_{r=1}^{|\mathbf{C}|} \exp(-\rho_r)) + v\mathbf{Tr}(\mathbf{M}) \\ \text{subject to} \quad & \mathbf{M} \succeq 0. \end{aligned} \qquad (4)$$

Here v is a regularizing parameter. The regularizer term prevents the optimization problem leading to learning an arbitrarily large M to make the exponential loss zero. Note the in the boosting technique distance metric M is learned by incrementally adding the rank-one and trace-one matrices U_k as bases learners. The crucial part of solving the optimization problem 4 is estimating all basis learners U_k and the number of such possible $U's$ is infinite. Therefore, instead of directly solving the primal problem 4, the column generation technique [5] is used, which finds the approximate solution by estimating the subset of entire variables set at each step. The new variable is learned incrementally only if it improves the solution significantly. The details of the algorithm can be found in [17].

IV. EXPERIMENTS

We conduct various experiments to evaluate the accuracy and effectiveness of our algorithm. Firstly, we demonstrate how the defined CQFB feature and the learned embedding improve the performance of texture classification tasks. In each case, we compare the performance to those alternatives from ablation studies or prior work. To ensure the comparability of different methods, the same test data is used for all tested methods. Our implementations are in Matlab, and we used the reference code from authors' websites as appropriate. Our data are drawn from the TUM texture repository [22]. The dataset has acceleration data corresponding to 69 different surface materials. The recording of training and testing data is done separately by different users. The training data is recorded by domain experts, whereas testing data is recorded by subjects unfamiliar with work details.

Each training and testing data contains 690 samples, ten samples in each class. In the prior work [22], the algorithm is evaluated using 69 different surface materials. The detail of the data recording procedure is described in [22]. We treat the samples within a class as equivalent, and randomly pick a representative every time we need features for a class. For all experiments below, testing is done on a separately recorded testing dataset, which is different from the training dataset.

Feature	Accuracy w/o boosting (%)	Accuracy with boosting (%)	Ref.
Spectral Peak	12	12	[10]
Spectral frequency	2	4	[10]
Energy Values	43	65	[16]
CQFB($\alpha=1$)	2.5	13	Ours
CQFB($\alpha=1.8$)	**60**	**82**	Ours

TABLE I: Comparison of k-NN classification performance with different features. The results show the improvement in classification accuracy for different features when the boosted embedding is used in conjunction.

Feature	Data	Dim	Bayes	Decision Tree	Ref.
LPC	Acc	10	5	4	[4]
Spectral peaks	Acc	5	9	9	[10]
Roughness	Acc	1	11	10	[7]
Energy	Acc	30	29	30	[16]
Spectrogram	Acc	125	39	29	[18]
Multimodal features	Acc, Img Snd, Frc	6	**74**	67	[22]
CQFB($\alpha=1.8$)	Acc	11	72	**69**	Ours

TABLE II: Comparison of classification accuracy (%) of our approach and related work by Strese *et al.* [22]. The results of related work are taken from the corresponding paper [22].[1]

Classification results: This section evaluates how effective the boosting-based feature transformation technique is in improving the classification performance of haptic textures. Training signals are labeled by their class, and the training triplets are constructed with two samples from a single class and the third sample from a different one. We seek to show that classes of signals are easily separable in the embedding space and hence perform a better classification task: predicting which of the 69 classes an unlabeled test signal belongs to. From a total of 690 training samples, we generated a total of 6210 triplets and trained metric M with a regularizing parameter $v = 10^{-7}$ in Eq. 4, for 3000 iterations.

Our work rests on two key components: (a) the CQFB(Constant Q-factor bandwidth) features that capture salient features of the acceleration data, and (b) the boosting-based feature transform that further enhances separability of signals in the projected space. To demonstrate the individual role of each of these factors to the overall efficacy of the method, we conduct various ablation studies in which we choose different features, or omit the metric embedding, or both. The results for the classification of unseen test signals

<hr>

[1]Some abbreviation used are as follows: Acc (acceleration), Img (Image), Snd (sound), Frc (friction), CQFB(Constant Q-factor filter bank).

with different features are given in Table I. In the first experimentation, Our boosted features use only a simple k-NN ($k=3$) classifier, to highlight the advantages of the features alone. The classification performance of various other classifiers is presented later in the section.

Table I shows the comparison of k-NN classification performance with different features from prior work, in original feature space, and in the learned boosted embedding space. Experimentation is done for various feature dimensions, and the best accuracy is reported for each feature in Table I. As we can see, some of the features such as spectral peak, spectral frequency cannot capture the hidden features of the data well. Note that the CQFB feature with constant bandwidth ($\alpha=1$) also fails to capture the complexity of underlying data. This is expected, as the discriminating power is better at a lower frequency, and by using an equal bandwidth filter, we lose signal feature by capturing lower frequency components at a low resolution. Using a variable bandwidth CQFB feature, accuracy improves significantly (from 2.5% to 60%). This further validates the significance of using CQFB feature in the classification task. As we can see from Table I, the learned metric consistently improves k-NN classification accuracy for most of the features, and the best accuracy is achieved with the combination of the CQFB feature and boosted embedding.

We also present accuracies with a variety of other features and classification methods from prior work. Table II reports various other classifier's results and its comparison from related work by Strese *et al.* [22]. In [22], the authors present the classification results of the Naive Bayes classifier and Decision Tree with several features. Note that all unimodal features which are obtained from acceleration data alone have low accuracy ($\leq 39\%$), and the accuracy improves to 74% for Naive Bayes classifier and 67% for Decision Tree only when other modalities of data are used for features extraction. In contrast, CQFB feature augmented by boosted embedding technique yields promising results only with acceleration data and is beaten marginally only by the method that resorts to other modalities to improve recognition.

Further, we compare in Fig. 2 the performance of the learned distance metric with the baseline Euclidean distance metric for classification tasks across various features dimensions using the CQFB feature. It can be observed that the boosted embedding improves the accuracy of the KNN classifier significantly. In Fig. 2b we study the same for other types of classifiers that one may select. As we can see in Fig. 2b, the classification performance of all classifiers improves in the embedding space. This validates that the learned distance metric is much more effective in improving the separability of data than the Euclidean metric.

Clustering: We visualize the embedding space (projected to 2D using t-SNE [13]) in Figure 3, with just randomly chosen ten classes out of 69 for clarity. On the left (a), we see constant bandwidth CQFB features ($\alpha=1$) fail to recover the underlying data structure. In the middle (b), variable bandwidth CQFB features form clear class-based
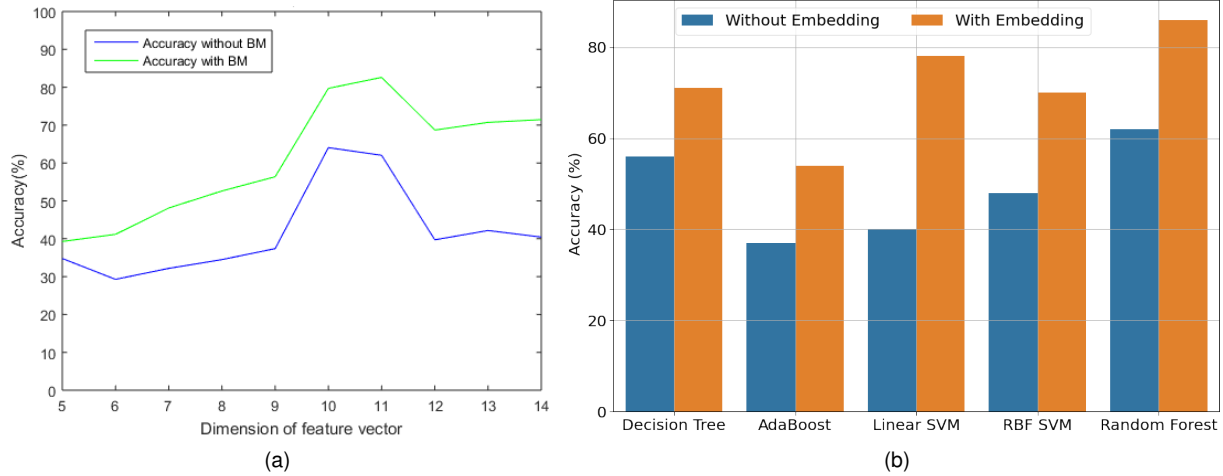
(a)

(b)

Fig. 2: Performance of k-NN classifier in Euclidean and boosted embedding space as a function of the feature dimension. The second figure shows the performance gain in classification accuracy in embedded space over Euclidean space for different classifiers.
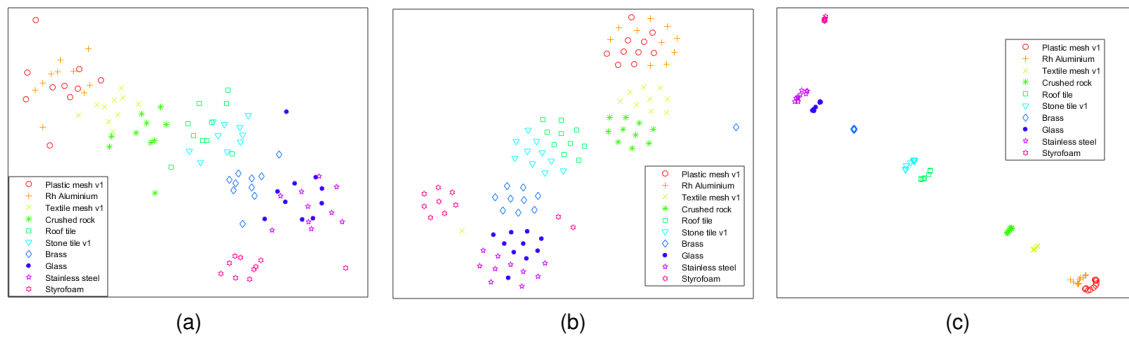

(a)

(b)

(c)

Fig. 3: Illustrating improved separability of classes due to the choice of CQFB feature and boosted embedding. Different symbols and colors indicate different materials. Features are projected onto 2D space using t-SNE plot, (a) Constant bandwidth feature (CQFB features with $\alpha = 1$), (b) Variable bandwidth feature (CQFB features with $\alpha = 1.8$) and (c) Variable bandwidth feature (CQFB features with $\alpha = 1.8$) projected after embedding.

clusters, but they are fairly diffused, and it is easy to see how a simple classifier can confuse similar classes near their class boundaries. On the right (c), boosting-based feature transformation induces more compact clusters with low intra-class variance, but high inter-class separation. Classes are much better separable in Fig. 3c.

Haptic discrimination: In this section, we evaluate the pairwise distinguishability of surface materials in the embedding space. We measure the pairwise distinguishability of two surface materials as the average number of triplet (x_i, x_j, x_k) comparisons for which the distance between intra-class samples x_i and x_j is lesser than the distance between inter-class samples x_i and x_k $(d_M(x_i, x_j) < d_M(x_i, x_k))$. We define the dissimilarity index p_{ab} for each material pair (a, b) as follows:

$$p_{bc} = \frac{|S|}{|C_{bc}|}; S = \{(x_i, x_j, x_k) | d_M(x_i, x_j) < d_M(x_i, x_k)\} \quad (5)$$

where (x_i, x_j, x_k) is a triplet constructed from two signals x_i and x_j of same class b, and x_k from different class c and C_{bc} is a set of all possible triplets constructed from signals

of class b and c without any constraint on the distance. The dissimilarity index (p_{bc}) is a measure of the probability of distinguishing two materials in the embedded space using the learned distance matrix M. A small value of the dissimilarity index indicates a more confusing material pair. Figure 4a and 4b shows the comparison between confusion matrices obtained using Euclidean and boosting with CQFB features. The darker spots in the confusion matrix indicate more confusion in distinguishing texture pairs. As we can see, materials are better separable in the learned embedding space than in the Euclidean space. The overall discrimination error is obtained by finding the mean of the confusion index of all material pairs. The average discrimination error in separating a material-pair using Euclidean metric and Mahalanobis metric are 7.10% and 3.76%, respectively, substantiating the usefulness of boosted embedding in improving separability of two classes of materials.

Limitation: We evaluate our model by comparing its performance with actual human subject performance in discriminating haptic textures. For human perceptual
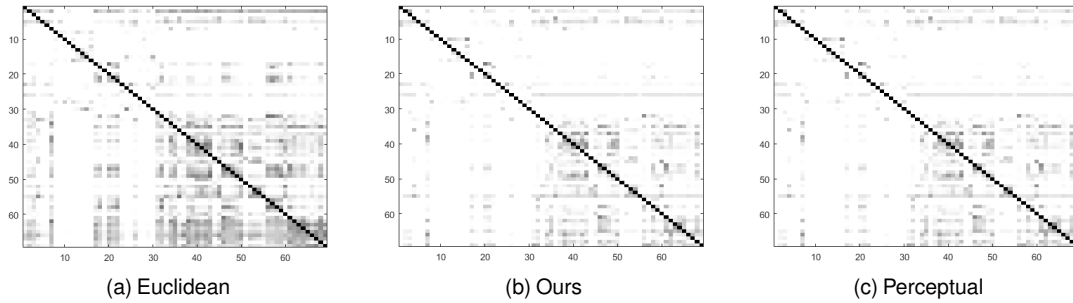
| (a) Euclidean | (b) Ours | (c) Perceptual |

Fig. 4: Confusion matrix (69×69 matrix) while discriminating different texture pairs obtained with (a) Euclidean metric, (b) Boosting-based Mahalanobis metric, c) Ground-truth perceptual distance obtained from human perceptual judgments [22]. Darker spots indicate more confusion in distinguishing a pair of textures. The matrix shows many indistinguishable textures pairs in the Euclidean space. Boosting improves the separability of surface textures. Many texture pairs (such as brick (index = 8) and granite (index = 10)) that are difficult to discriminate for users are readily distinguishable by the learned metric, while some texture pairs (plastic mesh (index = 4) and carpet (index = 36)) that are easy to discriminate by humans were more challenging for the learned metric.

similarity feedback, we have used the similarity matrix $\mathbf{S_H}$ provided by authors in [22]. Each element $\mathbf{S_H}(i,j)$ denotes the number of subjects (out of 30) having perceived different materials i and j to be similar and clubbed them into the same group. In other words, each element in the matrix $\mathbf{S_H}$ indicates the confusion in separating two materials apart. As we can see in Figure 4b and 4c, the dissimilarity trend in both semantic and perceptual distance matrices is quite different, indicating the semantic metric's inability to represent perceptual dissimilarity. We believe this is due to not incorporating human perceptual limitation in distinguishing two very similar signals (called just noticeable difference-JND) in the modeling process. The inclusion of human perceptual limitations in terms of JND in the distance metric learning process will definitely increase its usability.

V. CONCLUSIONS

In this work, we have presented a framework for the discriminative analysis of haptic textures using a learned distance metric. We demonstrated the effectiveness of hand-crafted spectral features in improving the distinguishability of real-world surface textures. Further, we observed that the learned boosted embedding on the CQFB space using class-based supervision improves separability between haptic signals. We found that our method outperforms the state-of-the-art method by a large margin in classification tasks when only the single-exploration acceleration data is used. The extensive experimentation with different predictors and feature dimensions demonstrates the effectiveness of our approach for several downstream tasks such as texture recognition and discrimination. Our current work mainly looks into class-based separability between objects. It will be intriguing to study how effectively the boosting-based technique can model human perceptual limitations such as JND.

REFERENCES

[1] S. Bensmaïa and M. Hollins. Pacinian representations of fine surface texture. *Perception & psychophysics*, 2005.
[2] K. Beyer, J. Goldstein, et al. When is "nearest neighbor" meaningful? In *ICDT*. 1999.
[3] D. S. Chathuranga, Z. Wang, et al. A biomimetic soft fingertip applicable to haptic feedback systems for texture identification. In *HAVE*. 2013.
[4] H. Culbertson, J. Unwin, et al. Modeling and rendering realistic textures from unconstrained tool-surface interactions. *ToH*, 2014.
[5] A. Demiriz, K. P. Bennett, et al. Linear programming boosting via column generation. *Machine Learning*, 2002.
[6] M. Enriquez, K. MacLean, et al. Haptic phonemes: basic building blocks of haptic communication. In *ICMI*. 2006.
[7] J. A. Fishel and G. E. Loeb. Bayesian exploration for intelligent identification of textures. *Frontiers in neurorobotics*, 2012.
[8] Y. Gao, L. A. Hendricks, et al. Deep learning for tactile understanding from visual and haptic data. In *ICRA*. 2016.
[9] J. Goldberger, G. E. Hinton, et al. Neighbourhood components analysis. In *NeurIPS*. 2005.
[10] N. Jamali and C. Sammut. Majority voting: Material classification by tactile sensing using surface texture. *ToR*, 2011.
[11] N. Landin, J. M. Romano, et al. Dimensional reduction of high-frequency accelerations for haptic rendering. In *EuroHaptics*. 2010.
[12] S. J. Lederman and R. L. Klatzky. Haptic classification of common objects: Knowledge-driven exploration. *Cognitive psychology*, 1990.
[13] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 2008.
[14] K. Priyadarshini, S. Chaudhuri, et al. Perceptnet: Learning perceptual similarity of haptic textures in presence of unorderable triplets. In *WHC*. 2019.
[15] B. A. Richardson and K. J. Kuchenbecker. Improving haptic adjective recognition with unsupervised feature learning. In *ICRA*. 2019.
[16] J. M. Romano and K. J. Kuchenbecker. Methods for robotic tool-mediated haptic surface recognition. In *Haptics Symposium*. 2014.
[17] C. Shen, J. Kim, et al. Positive semidefinite metric learning with boosting. In *NeurIPS*. 2009.
[18] J. Sinapov, V. Sukhoy, et al. Vibrotactile recognition and categorization of surfaces by a humanoid robot. *ToR*, 2011.
[19] L. Skedung, M. Arvidsson, et al. Feeling small: exploring the tactile perception limits. *Scientific reports,Nature*, 2013.
[20] M. Strese, L. Brudermueller, et al. Haptic material analysis and classification inspired by human exploratory procedures. *ToH*, 2019.
[21] M. Strese, C. Schuwerk, et al. Surface classification using acceleration signals recorded during human freehand movement. In *WHC*. 2015.
[22] —. Multimodal feature-based surface material classification. *ToH*, 2017.
[23] L. Torresani and K.-c. Lee. Large margin component analysis. In *NeurIPS*. 2007.
[24] E. P. Xing, M. I. Jordan, et al. Distance metric learning with application to clustering with side-information. In *NeurIPS*. 2003.
[25] H. Zheng, L. Fang, et al. Deep learning for surface material classification using haptic and visual information. *ToM*, 2016.