Handling Scan-time Parameters in Haptic Surface Classification

Alex Burka[1] and Katherine J. Kuchenbecker[2]

***Abstract*— The physical sensations of touch, as measured by the force and vibration felt during contact, strongly depend on the normal force exerted by the end-effector as well as its speed relative to the surface, factors we call "scan-time parameters." When researchers record surface interactions for machine learning tasks such as haptic surface recognition, they must either (1) precisely control these parameters, (2) record them alongside the rest of the data, or (3) develop post-processing that makes surface percepts invariant to scan-time parameters. Here we use multi-class support vector machines to compare the second approach ("scan-dependent features" developed in our prior work) with the third approach ("scan-free features" developed by Strese et al.). We first verify our implementation of the scan-free features by testing on Strese et al.'s published dataset of 69 surfaces, achieving 87.4% cross-validation accuracy. We then compare the two approaches on new surface interaction data gathered from 28 surfaces with our instrument (the Proton Pack), obtaining 57.9% accuracy using scan-dependent features and 93.3% accuracy using scan-free features, demonstrating superiority of the scan-free features for this task. We further calculate the correlation of all features to scan-time parameters, confirming that the scan-free features are relatively invariant to scan-time parameters.**

I. INTRODUCTION

When a human or a robot touches a surface through a tool, the resulting haptic sensations are highly dependent on normal force and tip speed, factors we call the *scan-time parameters*. In other words, an agent will feel different tangential forces and high-frequency vibrations depending on how they choose to touch a surface [1]. In order to build a good haptic model of a surface, e.g., for identification of newly encountered surfaces, one needs to control for these parameters in some fashion. An analogy can be drawn to the problem of controlling for illumination in computer vision. The same physical object will appear differently to a camera based on the ambient lighting of a scene, yet it is desirable to make models that are independent of this constantly varying parameter. Strategies for abstracting over brightness include controlling the lighting during model learning and evaluation (e.g., with a portable lighting rig), measuring scene brightness and including it in the object model (e.g., approximating the object's bidirectional reflectance distribution function (BRDF)), or formulating features that are invariant to lighting conditions (e.g., shape).

A similar choice must be made in haptic surface perception. First, we could address the problem by building a robotic armature to make repeatable motions, thereby dialing in a predetermined set of scan-time parameters, and build models for use only in such controlled settings. This approach is not very generalizable, increases the cost and complexity of the system, and presents problems in dissociating internal haptic sensations of the robotic armature from the surface interaction sensations that we want to measure [2]. Thus we will not consider it further here. A second possible method, which we have used in earlier work, is to measure the scan-time parameters during training and include them in the model, so their effects on the other features (*scan-dependent features*) can be characterized. Finally, the approach developed by Strese et al. [3], [4] is to engineer features that aim to be invariant to scan-time parameters; we call these *scan-free features*. This paper seeks to determine whether the second or third approach is superior.

II. BACKGROUND

The three approaches described above for controlling for scan-time parameters are all represented in the haptic sensing literature. For example, Fishel and Loeb [5] used a robotic armature to gather data while dragging a SynTouch BioTac at several predefined normal forces and tip speeds, obtaining classification accuracy of 95.4% over a dataset of 117 surfaces. However, this approach may not work well in less controlled settings. Our lab has typically used the second approach for classification. Our early work using the Haptic Camera, a pen-shaped device [6] that includes a force/torque sensor, accelerometers and magnetic tracking, showed that including scan-time parameters, friction forces, and specific transformations of the acceleration signal as model features improved surface classification from 30.6% to 72.4% on 15 surfaces [7].

Other notable prior instruments helped inspire our work with the Proton Pack. The WHaT [8], another pen-shaped device featuring accelerometers, is wireless but did not track its own motion until Andrews et al. added external visual tracking [9]. Xu et al. experimented with a Shadow Dextrous Hand making precisely controlled surface interactions through a SynTouch BioTac [10]. Finally, Strese et al. used a custom pen containing accelerometers, force-sensitive resistors, and a microphone, either held freehand or attached to a Phantom Omni device [3], [4].

This material is based upon work supported by the U.S. National Science Foundation (NSF) under grant 1426787 as part of the National Robotics Initiative. Alex Burka was also supported by the NSF under the Complex Scene Perception IGERT program, award number 0966142.

[1]A. Burka is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. aburka@seas.upenn.edu

[2]K. J. Kuchenbecker is with the Departments of Mechanical Engineering and Applied Mechanics, Computer and Information Science, and Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA and with the Haptic Intelligence Department, Max Planck Institute for Intelligent Systems, Stuttgart, BW 70569, Germany. kjk@is.mpg.de

In this work we will use surface features that depend on the scan-time parameters, which come from earlier work using the Haptic Camera [7] and the Proton Pack [11], [12], as well as features that were carefully engineered to be less correlated with scan-time parameters, based on the work of Strese et al. [3], [4]. The first of those two papers [3] achieved an extraordinarily high classification accuracy of 95% in cross-validation over a training set of dragging and tapping interactions with 69 surfaces, leading us to a direct comparison of the approaches.

A. Hardware

The Proton Pack is a portable, self-contained visuo-haptic surface recording device, integrating a camera, six-axis force/torque sensor, dual high-bandwidth two-axis accelerometers, and a tooling ball end-effector in a handheld rig. It uses computer vision to track its own motion relative to fixed fiducial markers. Development of the system can be traced from its introduction [11] through various improvements [12] in preparation for collecting a vast set of visual and haptic surface interaction data.

III. EXPERIMENTS

Three experiments were performed. First, we sought to replicate Strese et al.'s highly accurate classification results using their published data and features [3], [4]. Second, we added the scan-free features to our own surface models and evaluated classification performance on new data. Third, we examined the extent to which Strese et al.'s scan-free features are truly independent of the scan-time parameters.

A. Replication study

We downloaded the Lehrstuhl für Medientechnik (LMT) texture database [13] and preprocessed it using Strese et al.'s instructions [3], [4]. This dataset includes acceleration, friction and sound data recorded when experimenters tapped and dragged a rigid metal tool on 69 surfaces, divided into training and testing sets each containing ten scans per surface. Tangential force was estimated by measuring the force exerted on the tool handle by the experimenter with a differential pair of force-sensing resistors; this signal may be used to derive approximate friction characteristics, although the scan-time parameters of normal force and tip speed were not measured.

In Strese et al.'s latter paper [4], features are identified by acronyms with a two-letter prefix indicating the sensory modality (A for acceleration, F for friction, and S for sound) and the exploratory procedure (I for impact/tapping, and M for movement/dragging). We implemented all of the given acceleration, friction and sound features, but we excluded the image features to focus on haptic perception. A brief overview of the calculations is given below, largely summarized and reproduced from [3], [4] with a few clarifications and corrections [14], marked with a star ($\star$).

Relevant signals are a_l (acceleration leading up to tapping impact), a_i (acceleration during impact), a_m (acceleration during movement phase), and f_m (friction measurement during movement phase), all sampled at 10 kHz. The acceleration signals were reduced to a single axis using the DFT321 algorithm [15]. Many features depend on a sliding time window, denoted as $\text{win}(k) = a_k ... b_k$ where k starts at 1 and the kth window ranges from a_k to b_k. Additionally, several mathematical functions are repeatedly used: $\sigma(\text{signal})$, the standard deviation of a signal; $\text{lpf}(\text{signal}, f)$, which signifies a low-pass filter with cutoff frequency f; $\text{dct}(\text{signal}, m)$, the discrete cosine transform at a length of m samples; and $\text{sma}(\text{signal}, n)$, a simple moving average of width n samples.

*1) Modified melfrequency cepstral coefficients (*AMCC*) $\star$:* Strese et al. [4] define overlapping sliding windows of width 25 ms and separation 15 ms. The HTK-MFCC library [16] is used to extract coefficients from each window, and the average values of coefficients 2 through 14 are used as the first 13 features.

$$\begin{aligned} \text{win}(k) &= (15\,\text{ms})(k-1)\ ...\ (15\,\text{ms})(k-1) + 25\,\text{ms} \\ \vec{c}^{\,k} &= \text{mfcc}(a_m[\text{win}(k)]) \\ \text{AMCC} &= \frac{1}{N_w}\sum_{k=1}^{N_w}\left[\ c_2^k \mid ... \mid c_{14}^k\ \right] \end{aligned} \tag{1}$$

*2) Impact hardness (*AIH*):* Strese et al. [4] locate the three highest peaks of the acceleration during tapping impact and average their time indices n_1, n_2, n_3, measured from the time of impact. The hardness feature is the value of the highest peak divided by that average index, and additionally normalized by the sum of the smoothed acceleration during the lead-up to impact (which approximates the impact velocity).

$$\begin{aligned} \overline{n} &= \frac{n_1 + n_2 + n_3}{3} \\ a_{l,filt} &= \text{lpf}(a_l, 12\,\text{Hz}) \\ \text{AIH} &= \frac{\max|a_i|}{\overline{n}}\frac{1}{\sum a_{l,filt}} \end{aligned} \tag{2}$$

*3) Impact spectral centroid (*AISC*):* Strese et al. [4] define spectral centroid to be the average of the squared discrete cosine transform (DCT) of the acceleration during impact, weighted by the DCT frequencies.

$$\begin{aligned} I &= \text{dct}(a_i, 4096) \\ \text{AISC} &= \frac{\sum_{k=1}^{\frac{m}{2}} I(f_k)^2 f_k}{\sum_{k=1}^{\frac{m}{2}} I(f_k)^2} \end{aligned} \tag{3}$$

*4) Temporal roughness (*AMTR*):* Strese et al. [4] measure roughness in the time domain using the Coiflet3 wavelet transform of acceleration. d_1 and d_5 are the reconstructed wavelet coefficients at detail levels 1 and 5. The log mean difference between d_1 and a rescaled version of d_5 aims to capture microscopic surface features.

$$\begin{aligned} d_1\ ...\ d_5 &= \text{coiflet3}(a_m) \\ \text{AMTR} &= \log_{10}\left(\overline{d_1 - \frac{\overline{d}_1}{\overline{d}_5}d_5}\right) \end{aligned} \tag{4}$$

*5) Spectral roughness (*AMSR*) ⋆:* Strese et al. [4] evaluate roughness in the frequency domain by comparing changes in the windowed DCT over time.

$$\begin{aligned} \mathrm{win}(n) &= (0.5\,\mathrm{s})(n-1)\ \dots\ (0.5\,\mathrm{s})n \\ X_n &= \mathrm{dct}(a_m[\mathrm{win}(n)], 4096) \\ D_k &= X_n - X_{n+100} \\ \mathrm{AMSR} &= \log_{10}\left(\sum_{k=1}^{K} D_k^2\right) \end{aligned} \tag{5}$$

*6) Waviness (*AMWV*) ⋆:* Strese et al. [4] use the standard deviation of the difference between two differently filtered versions of the same signal to measure the "deviation of the low-frequency signal slopes".

$$\begin{aligned} \mathrm{win}(k) &= (20\,\mathrm{ms})(k-1)\ \dots\ (20\,\mathrm{ms})k \\ a_{m,filt} &= \mathrm{lpf}(a_m, 100\,\mathrm{Hz}) \\ m_k &= \overline{|a_{m,filt}[\mathrm{win}(k)]|} \\ \mathrm{AMWV} &= 1 + \log_{10}\left(\sigma(m - \mathrm{sma}(m, 100))\right) \end{aligned} \tag{6}$$

*7) Spikiness (*AMSP*):* Strese et al. [4] detect spikes using a very wide moving average to obtain an indicator of surfaces with spatially discontinuous features.

$$\begin{aligned} a_{m,filt} &= \mathrm{lpf}(a_m, 100\,\mathrm{Hz}) \\ x_{sma} &= \mathrm{sma}(a_{m,filt}, 5000) \\ x_{th} &= 2\sigma(a_{m,filt}) + \overline{a_{m,filt}} - \overline{x}_{sma} \\ \mathrm{AMSP} &= \log_{10}\left(\overline{x - x_{th}}\right) \end{aligned} \tag{7}$$

*8) Fineness (*AMF*) ⋆:* Strese et al. [4] repeat the spectral centroid calculation on the dragging portion of the acceleration signal, averaged over one-second windows.

$$\begin{aligned} \mathrm{win}(q) &= (1\,\mathrm{s})(q-1)\ \dots\ (1\,\mathrm{s})q \\ I_q &= \mathrm{dct}(a_m[\mathrm{win}(q)], 4096) \\ f_q &= \frac{\sum_{k=1}^{\frac{m}{2}} I_q(f_k)^2 f_k}{\sum_{k=1}^{\frac{m}{2}} I_q(f_k)^2} \\ \mathrm{AMF} &= \frac{1}{Q}\sum_{q=1}^{Q} f_q \end{aligned} \tag{8}$$

*9) Regularity (*AMRG*):* Strese et al. [4] state that autocorrelation of the acceleration signal can reveal repetitive patterns on the surface.

$$\begin{aligned} \hat{x} &= \frac{a_m}{\max a_m - \min a_m} \\ r_k &= \mathrm{xcorr}(\hat{x}) \\ \mathrm{AMRG} &= \overline{\Delta r_k} \end{aligned} \tag{9}$$

*10) Unnormalized friction (*Fr*):* This feature, only used in the earlier paper by Strese et al. [3], is simply an average of the tangential force.

$$\mathrm{Fr} = \overline{f_m} \tag{10}$$

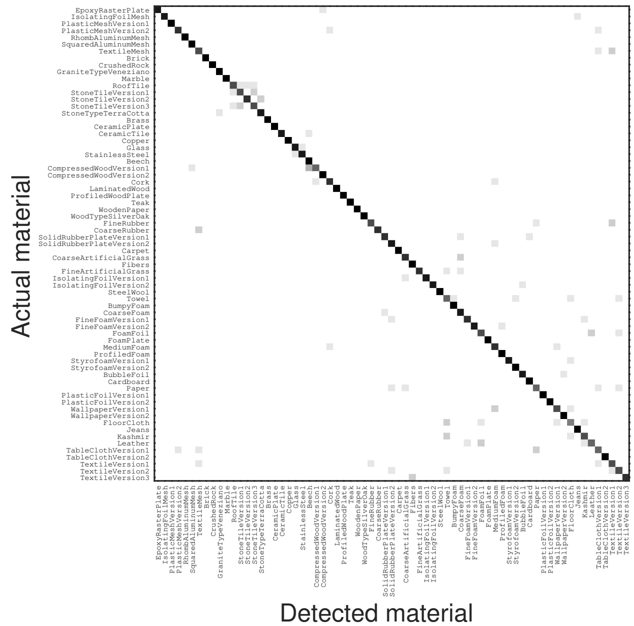


Fig. 1: Confusion matrix for SVM classification on the LMT texture database using scan-free features. The trained model, which obtains 87.4% overall accuracy, is a ν-SVM with a radial basis function (RBF) kernel, $\nu = 0.1, \gamma = 0.5$.

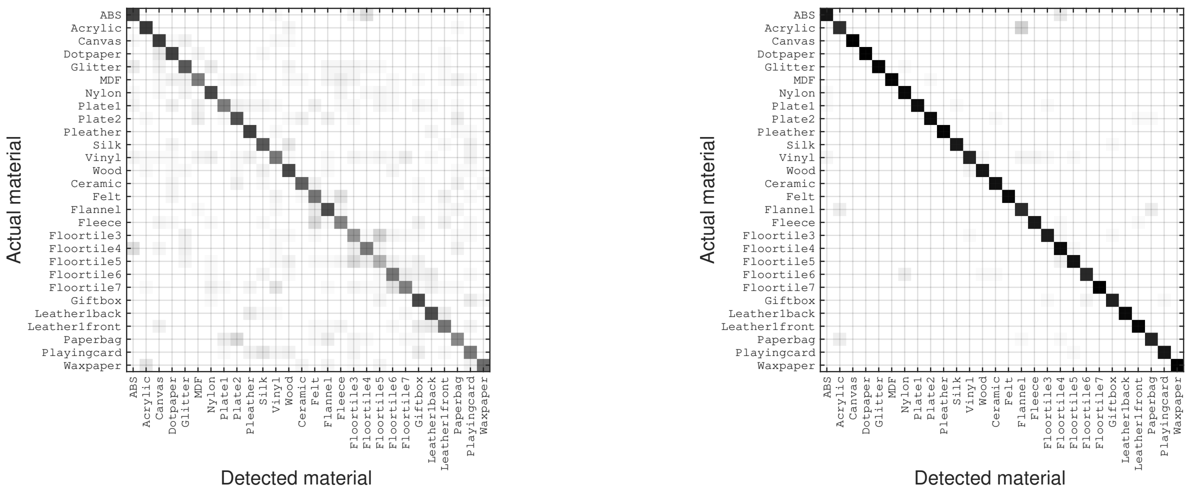
*11) Normalized friction (*FM*):* This feature, which Strese et al. introduced in their latter paper [4], improves upon Fr by putting the average acceleration in the denominator, to try to correct for the dependence between normal and tangential force.

$$\mathrm{FM} = \frac{\overline{|f_m|} + \sigma(\Delta|f_m|)}{\overline{|a_m|}} \tag{11}$$

We also implemented the sound features from the latter Strese et al. paper [4], but we omit their formulae here to focus on haptic features.

The provided training set contains ten $25\,\mathrm{s}$ recordings of acceleration and sound for each surface (including tapping and dragging), plus $100\,\mathrm{s}$ of friction data (including only dragging, recorded separately). We divided the friction data into $10\,\mathrm{s}$ chunks and associated one with each of the acceleration/sound trials, even though it was not recorded simultaneously. All of the above features were calculated and then standardized by extracting the mean and range of each feature over the training set, then subtracting that mean from both the training and validation features, and dividing by the range.

We trained support vector machines for classification using ten-fold cross-validation (in each fold, nine recordings were used for training, and the remaining one for validation). We observed 87.4% average classification accuracy on the training set. The confusion matrix representing the performance of this classifier is shown in Fig. 1. Although somewhat lower than the 95% accuracy value reached by Strese et al. [3], this value is comparable. Some discrepancy here is expected and explained by the presumably different cross-validation



(a) Scan-dependent only, 57.9% accuracy
10 linear bins, $\alpha = 0.2, \nu = 0.15, \gamma = 10$

(b) Scan-free only, 90.4% accuracy
$\nu = 0.15, \gamma = 1$

Fig. 2: Confusion matrices for classifiers using scan-dependent and scan-free features only, at a $0.25\,\mathrm{s}$ window length.

processes, as well as the differences in the details of feature selection and machine learning algorithms employed. This successful replication validates our implementation of the scan-free features.

B. Improving classification with scan-free parameters

We built on our group's previous surface classification work [11], [12] by adding the scan-free features invented by Strese et al. [3], [4] to our feature vector.

A dataset was collected using the Proton Pack, containing approximately $20\,\mathrm{s}$ of pose, force, and acceleration data while a human operator dragged the tooling ball end-effector back and forth across each surface at various speeds (up to about $20\,\mathrm{cm/s}$) and normal forces (up to about $25\,\mathrm{N}$). The surfaces are 28 10-cm-square surface patches from the Penn Haptic Texture Toolkit [1]. Following the process in our previous publications [11], [12] we divided the dragging portions of these surface recordings into windows and calculated a feature vector for each window. We tried three window lengths: $0.05\,\mathrm{s}$, $0.25\,\mathrm{s}$, and $1.25\,\mathrm{s}$. The dragging portion was automatically segmented by locating the region of the recording in which the vertical position of the contact point was within $6\,\mathrm{mm}$ above its median position.

For this experiment we used a feature vector consisting of the power of the acceleration in regularly spaced frequency bins; mean and standard deviation of normal force, tangential force and tip speed; and the AMCC, AMTR, AMSR, AMWV, AMSP, AMF, AMRG, Fr, and FM scan-free features, defined in Section III-A. These scan-free features were selected because they operate on the same data as the extant scan-dependent features: acceleration and force data during the dragging phase (tapping was not considered in this experiment). At each window size, we divided the data equally into five folds. Holding out each fold as a test set in turn, we trained a support vector machine (ν-SVM) with a radial basis function (RBF) kernel on the remaining four folds. Grid search over the hyperparameters of the SVM (learning parameter ν and kernel width γ) and the scan-dependent features (number, shape, and width of acceleration frequency bins), evaluated using three-fold cross-validation, was used to find the optimal configuration. Table I shows the results achieved by the SVM for each window width and set of features. Figs. 2a and 2b show confusion matrices for classification using a $0.25\,\mathrm{s}$ window and, respectively, scan-dependent and scan-free features. Though we have a relatively small set of surfaces, it is clear that the scan-free features are much more effective at classifying the surfaces. In particular, the scan-dependent classifier is weak in distinguishing between certain types of floor tiles and paper-like surfaces, in addition to having more misclassifications overall. Interestingly, the scan-free classifier's confusion between the flannel and acrylic surfaces is striking, but this may indicate that the classifier "sees through" the thin flannel to the rigid plastic backing.

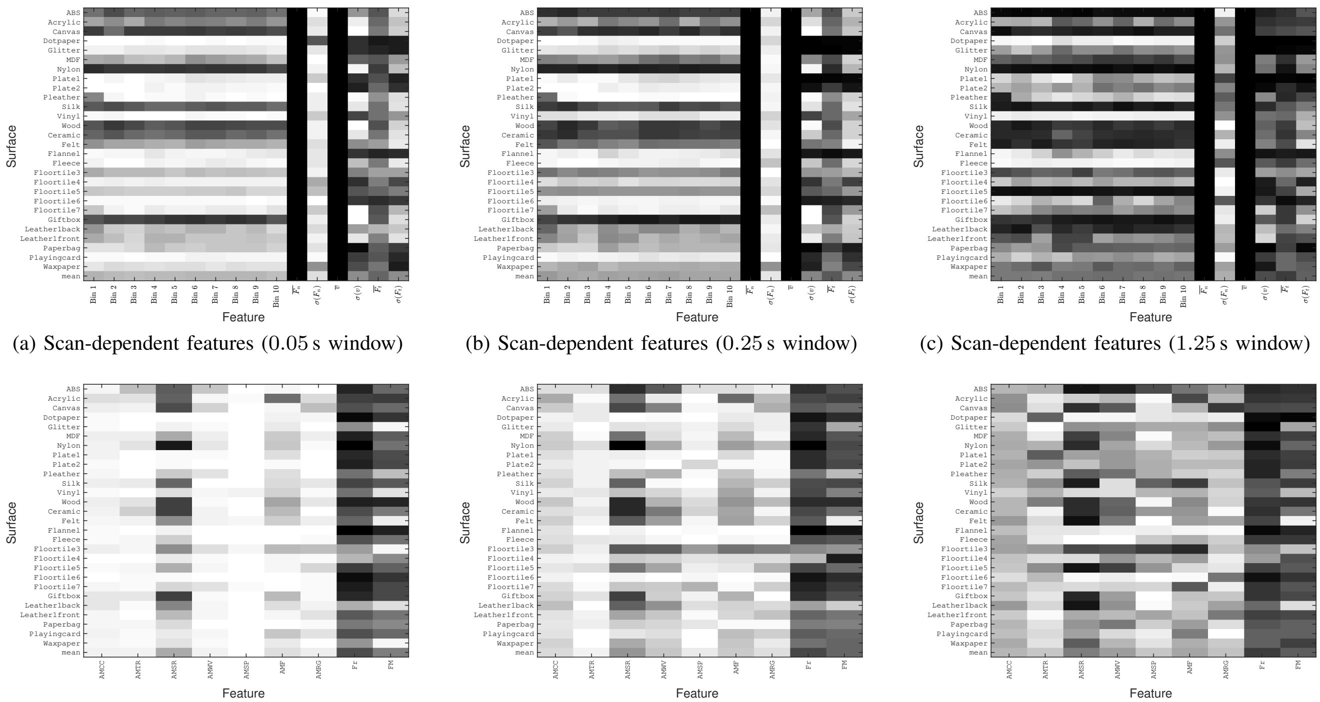
Window	Scan-dependent (%)	Scan-free (%)	All
$0.05\,\mathrm{s}$	$49.01\ (\sigma = 1.3)$	$56.99\ (\sigma = 1.5)$	$63.79\ (\sigma = 1.2)$
$0.25\,\mathrm{s}$	$59.92\ (\sigma = 2.6)$	$92.30\ (\sigma = 1.2)$	$92.91\ (\sigma = 0.9)$
$1.25\,\mathrm{s}$	$38.95\ (\sigma = 5.0)$	$97.37\ (\sigma = 1.9)$	$96.58\ (\sigma = 0.7)$

TABLE I: Classification accuracies with different feature sets. Chance performance would be 3.6% for 28 surfaces.

These results show that of the features considered here, the scan-free features of Strese et al. [3], [4] are clearly the better choice for haptic surface classification. However, the scan-free features are about two orders of magnitude slower to calculate in our implementation, so computational resources may be a relevant concern.

C. Parameter sensitivity

The development of scan-free features [3], [4] is predicated on the assumption that the engineered features do not depend on normal force or tip speed. However, previous work



(a) Scan-dependent features (0.05 s window)
(b) Scan-dependent features (0.25 s window)
(c) Scan-dependent features (1.25 s window)
(d) Scan-free features (0.05 s window)
(e) Scan-free features (0.25 s window)
(f) Scan-free features (1.25 s window)

Fig. 3: Feature sensitivity to scan-time parameters. Sensitivities are shown as the R^2 value mapped onto a scale from white (no correlation) to black (100% correlation). For AMCC, the average sensitivity of the cepstral coefficients is used. Note: Figs. 3a, 3b and 3c include the scan-time parameters ($\overline{F_n}$ and $\overline{v}$ columns), so those correlations are always 100%.

did not directly measure these parameters, so the assumption is unchecked. In our data, we do have those measurements, so we can check the assumption. Splitting the data into the same 0.05 s, 0.25 s, and 1.25 s windows, we recalculated the scan-dependent and scan-free features on each window.

For each surface and window, we can calculate the average normal force and scanning speed, as well as a value for each feature. We then attempt to fit a bilinear model relating each feature value to these average scan-time parameters. The fit coefficients are not important for our purposes, but the R^2 goodness-of-fit measure will be high when a feature strongly depends on scan-time parameters. The visualizations in Fig. 3 show the correlation between the scan-time parameters and the two types of features for each surface and window length, as well as average correlations across surfaces.

As expected, the scan-free features are much less dependent on the scan-time parameters in general. However, there are exceptions. In particular, the two friction features Fr and FM are relatively highly scan-dependent, which is consistent with the simplistic two-parameter model of friction as a tangential force linearly proportional in magnitude to normal force and tangential speed; adding a normalization based on vibration as in FM reduces but does not eliminate the correlation, most likely because vibration magnitude highly depends on tip speed as well [17]. Also, some purportedly scan-free features are more scan-free than others: using a 0.25 s window, the average R^2 values imply that "spikiness" or AMSP is the least correlated with scan-time parameters, while "spectral roughness" or AMSR is the most (excluding the friction features). However, this pattern varies widely by surface, which underscores the importance of considering a representative training set of diverse surfaces.

Curiously, all features (scan-free and scan-dependent) appear to be slightly more correlated with the scan-time parameters at the longer window length, despite the fact that classification is more effective. A possible explanation is that shorter windows are more sensitive to random sensor noise, making features appear to be less correlated with scan-time parameters than they really are – but that random noise makes classification harder with short windows. Indeed, a purely random feature would be perfectly scan-free but would not enable accurate classification.

IV. CONCLUSION

We demonstrated that the scan-free features invented by Strese et al. [3], [4] are effective on the provided data and on data collected by our custom instrument, the Proton Pack. Furthermore, most of these features are, indeed, not highly correlated with scan-time parameters, which suggests that they can be used effectively in situations where the scan-time parameters are unknown.

The scan-free classification pipeline is highly effective in our surface classification experiment (Sec. III-B). Strese et al.'s friction features, which cannot truly measure friction without taking normal force into account, are fairly highly correlated with the scan-time parameters, so it may seem that including them allows information about the scan-time parameters to leak into the scan-free classification pipeline. However, this is not a problem: if the friction features are omitted entirely, the accuracies in Table I remain virtually unchanged. This stability implies that the scan-free acceleration features are most important to the superior classification performance. We speculate that this dominance stems from the highly nonlinear nature of the features, which reflects the nonlinearity of the physical interactions that generate haptic sensations during dragging, as well as the fact that using features inspired by audio engineering draws upon that discipline's decades of experience in identifying salient properties and useful transformations.

Further studies in the area of scan-free features should consider other possible feature sources, including surface appearance and shape, as well as scan-time parameters not considered here that could affect the data, e.g., probe angle.

Future work with the Proton Pack includes collecting more data, with the goal of publishing a vast multimodal surface interaction dataset. We will include scan-time parameters in our recorded data, but some consumers of the data may not have the equipment necessary to measure them effectively, so a viable demonstration of scan-free surface modeling is valuable. Furthermore, it may be possible to use such a database to engineer surface interaction features that depend even less on scan-time parameters.

We plan to move away from hand-engineered features in favor of using deep neural networks to build surface models, as begun by our collaborators Gao et al. [18], a process which will benefit from having a vast dataset available. Since a neural network is free to concoct internal representations of the input in any way that turns out to be useful, yet is limited to fairly simple calculations, it will be interesting to examine the dependence of these representations on scan-time parameters.

REFERENCES

[1] H. Culbertson, J. J. Lopez Delgado, and K. J. Kuchenbecker, "One hundred data-driven haptic texture models and open-source methods for rendering on 3D objects," in *Proc. IEEE Haptics Symp.*, 2014, pp. 319–325.

[2] W. McMahan and K. J. Kuchenbecker, "Spectral subtraction of robot motion noise for improved vibrotactile event detection," in *Haptics: Perception, Devices, Mobility, and Communication: Proc. EuroHaptics*, 2012, pp. 326–337.

[3] M. Strese, C. Schuwerk, and E. Steinbach, "Surface classification using acceleration signals recorded during human freehand movement," in *Proc. IEEE World Haptics Conf.*, 2015, pp. 214–219.

[4] M. Strese, C. Schuwerk, A. Iepure, and E. Steinbach, "Multimodal feature-based surface material classification," *IEEE Transactions on Haptics*, 2016, doi:10.1109/TOH.2016.2625787.

[5] J. A. Fishel and G. E. Loeb, "Bayesian exploration for intelligent identification of textures," *Frontiers in Neurorobotics*, vol. 6, pp. 1–20, June 2012.

[6] H. Culbertson, J. Unwin, B. E. Goodman, and K. J. Kuchenbecker, "Generating haptic texture models from unconstrained tool-surface interactions," in *Proc. IEEE World Haptics Conf.*, 2013, pp. 295–300.

[7] J. M. Romano and K. J. Kuchenbecker, "Methods for robotic tool-mediated haptic surface recognition," in *Proc. IEEE Haptics Symp.*, 2014, pp. 49–56.

[8] D. K. Pai and P. Rizun, "The WHaT: a wireless haptic texture sensor," in *Proc. IEEE Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 3–9.

[9] S. Andrews and J. Lang, "Interactive scanning of haptic textures and surface compliance," in *Proc. IEEE Int. Conf. on 3-D Digital Imaging and Modeling*, 2007, pp. 99–106.

[10] D. Xu, G. E. Loeb, and J. A. Fishel, "Tactile identification of objects using Bayesian exploration," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 3056–3061.

[11] A. Burka, S. Hu, S. Helgeson, S. Krishnan, Y. Gao, L. A. Hendricks, T. Darrell, and K. J. Kuchenbecker, "Proton: A visuo-haptic data acquisition system for robotic learning of surface properties," in *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration*, 2016.

[12] A. Burka, A. Rajvanshi, S. Allen, and K. J. Kuchenbecker, "Proton 2: Increasing the sensitivity and portability of a visuo-haptic surface interaction recorder," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2017, accepted.

[13] M. Strese, C. Schuwerk, R. Chaudhari, and E. Steinbach, "LMT haptic texture database," http://www.lmt.ei.tum.de/downloads/texture/, accessed: 2017-01-11.

[14] M. Strese, Personal correspondence, 2016–2017.

[15] N. Landin, J. M. Romano, W. McMahan, and K. J. Kuchenbecker, "Dimensional reduction of high-frequency accelerations for haptic rendering," in *Haptics: Generating and Perceiving Tangible Sensations, Proc. EuroHaptics*, 2010, pp. 79–86.

[16] K. Wojcicki, "HTK MFCC matlab," https://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab, accessed: 2017-01-11.

[17] H. Culbertson and K. J. Kuchenbecker, "Should haptic texture vibrations respond to user force and speed?" in *Proc. IEEE World Haptics Conf.*, 2015, pp. 106–112.

[18] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 536–543.