2016 IEEE International Conference on Multisensor Fusion
and Integration for Intelligent Systems (MFI)
Kongresshaus Baden-Baden, Germany, Sep. 19-21, 2016

Proton: A Visuo-Haptic Data Acquisition System for Robotic Learning of Surface Properties

Alex Burka[1], Siyao Hu[1], Stuart Helgeson[1], Shweta Krishnan[1], Yang Gao[2], Lisa Anne Hendricks[2], Trevor Darrell[2], and Katherine J. Kuchenbecker[1]

***Abstract*— Autonomous robots need to efficiently walk over varied surfaces and grasp diverse objects. We hypothesize that the association between how such surfaces look and how they physically react during contact can be learned from a database of matched haptic and visual data recorded from various end-effectors' interactions with thousands of real-world surfaces, such as wood flooring, upholstered fabric, asphalt, grass, and anodized aluminum. As the first step in this effort, we detail the design and construction of the Proton, a multimodal data acquisition system that a human operator can use to gather the envisioned data set. Its sensory modalities include RGBD vision, egomotion, contact force, and contact vibration. Three interchangeable end-effectors (a SynTouch BioTac artificial fingertip, an OptoForce three-axis force sensor, and a steel tooling ball) allow for different material properties at the contact point and provide additional tactile data. This sensor suite emulates the capabilities of the human senses of vision and touch, with the goal of learning surface classification methods that are robust over different sensory modalities. We detail the calibration process for the motion and force sensing systems, as well as a proof-of-concept surface discrimination experiment using the tooling ball end-effector and a Vicon motion tracker. A multi-class SVM trained on the collected force and vibration data achieved 82.5% classification accuracy among five sample surfaces.**

I. MOTIVATION

Robots are being deployed in many unstructured environments that require rich sensing and thorough understanding of the surroundings, such as homes and hospitals. Haptic surface perception is critical for such robots to select locomotion strategies on different surfaces or choose how to grasp unfamiliar objects. For example, the 2015 DARPA Robotics Challenge [1] required robots to walk on and grasp diverse surfaces; many participants had trouble with friction perception and interpretation of contact force readings [2]. Haptic sensing can help a robot correct errors when trying out a gait or a grasp, but such interactions would become simpler if haptic properties could be accurately predicted from visual data before contact.

How can we give robots the ability to anticipate mechanical surface properties via vision? While each individual robot could learn such associations on its own, we envision a more efficient solution that does not require all robots to have sophisticated haptic sensors. Robots ought to be able to pool their perceptions using individual sensors and use each others' knowledge to improve their predictions. Before this goal can be reached, though, we need new methods to correlate haptic features across different sensory modalities.

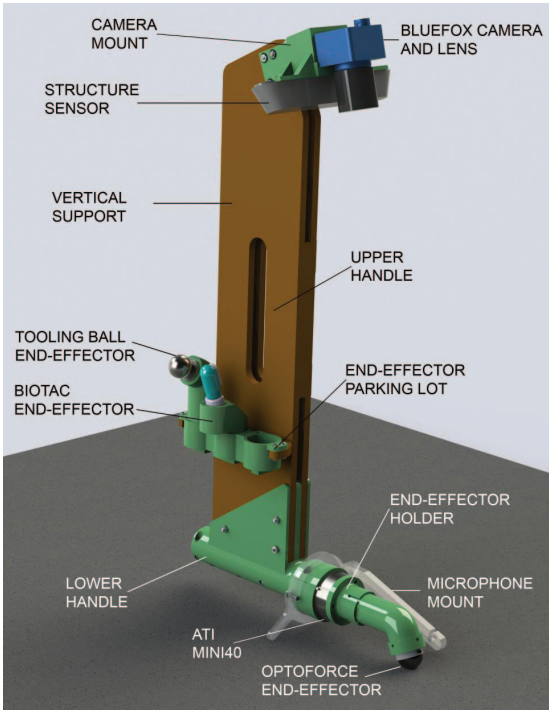

Fig. 1. A labeled CAD model of the Proton, shown with the OptoForce end-effector contacting a surface.

Almost every robot has a camera, but comparatively few have haptic sensing. Even when present, haptic sensors are diverse in modality and fidelity, and their outputs tend to drift over time due to thermal fluctuations and wear and tear. Consequently, the best feature space for representing how surfaces feel is still an open question. Additionally, there are few large public repositories of haptic surface interaction data. This project's long-term goal is to solve these issues by collecting a large dataset of multimodal surface data, including matched visual and haptic data labeled with human ratings of surface properties.

This paper presents the PROTONPACK (Portable Robotic Optical/Tactile ObservatioN PACKage, or Proton for short), a new portable visuo-haptic sensing rig (Fig. 1). After surveying prior work (Sec. II), we detail our design decisions and tradeoffs (Sec. III), hardware and software implementation (Sec. IV), system calibration (Sec. V), preliminary results from classifying a set of five surfaces (Sec. VI), and conclusions (Sec. VII).

This material is based upon work supported by the U.S. National Science Foundation (NSF) under grants 1426787 and 1427425 as part of the National Robotics Initiative. Alex Burka is also supported by the NSF under the Complex Scene Perception IGERT program, via grant 0966142.

[1]MEAM, ESE, and CIS Depts., GRASP Lab, University of Pennsylvania
{aburka,siyaohu,kuchenbe}@seas.upenn.edu
[2]CS Division, University of California, Berkeley
trevor@eecs.berkeley.edu

II. LITERATURE REVIEW

Stansfield [3] was one of the first to try to adapt the principles inherent to the human tactile system to create a robotic haptic perception system. Lederman and Klatzky had just discovered that humans experience and identify surfaces via "exploratory procedures" [4], specific movements that elucidate the desired haptic property of an object. For example, pressing into an object reveals its softness, while moving one's finger along a surface shows its texture. Stansfield programmed a PUMA 560 robot arm equipped with a 160-element tactile array and a force/torque sensor to physically interact with objects [3].

Following Stansfield, several haptic sensing systems have been developed for autonomous material or texture classification. These systems use many kinds of haptic sensors, such as an ATI Nano17 force/torque sensor [5], high-bandwidth accelerometers [5], [6], and custom artificial fingers with strain gauges or PVDF sensing elements [7], [8]. Other researchers employ the commercially available SynTouch BioTac biomimetic tactile sensor [9], [10]. As established by Lederman, Klatzky, and colleagues [11], changing the speed or force of the contact interaction typically alters the amplitude and frequency of the resulting haptic signals. Consequently, these haptic sensing projects have all carefully controlled the physical interactions between the sensing system and the surface using an upper-torso humanoid [5], [6], a robotic arm [8], or a custom test rig [9].

More recent work in this domain focuses on active perception. For example, Xu et al. [12] used a robot-mounted BioTac to classify objects, selecting exploratory movements online for maximum information gain. Similarly, Lepora et al. [13] compared the performance of passive and active perception under uncertainty, showing that active touch can compensate for noisy sensors. Having a robot wield the sensing system using a set of pre-programmed motions typically yields consistent contact force and scanning movements, and therefore repeatable haptic signals. However, significant effort must be expended to program the interaction controller to function well for a wide variety of surface properties, which are unknown at the start. One must also ensure that the motion of the robot itself does not generate significant tactile vibrations, as they may obscure the sensations caused by contact [14].

The other major approach to capturing haptic surface data is to use a handheld sensing system wielded by a human experimenter. For example, Pai and Rizun created the WHaT, a wireless haptic texture sensor that includes two miniature two-axis accelerometers and a piezoresistive force sensor in a pen-shaped shell [15]. This design allows measurement of natural surface interactions through a probe at the tip. Being wireless, this sensing system has a large workspace; however, one key disadvantage of this design is that the device cannot track its position or orientation. To solve this problem, Andrews et al. [16] added optical motion tracking and a linear Kalman filter to compensate for noise. This system was used to measure both stochastic and patterned textures, and to estimate surface compliance.

The ThimbleSense haptic measurement device [17] is worn on a human or robotic fingertip and includes an ATI Nano17 six-axis force/torque sensor within a thin shell. An external Vicon system tracks the position of each fingertip. Another force-sensor-based haptic capture system is that of Höver et al. [18]; a human operator explores real objects using a PHANToM Desktop haptic interface that is equipped with a custom probing tool and an ATI Nano17. The captured force data can be interpolated and used to model arbitrary non-linear materials with visco-elastic behavior. Finally, Culbertson and colleagues created a haptic recording device similar to the one presented in this paper. Their device is small and pen-shaped, containing an ATI Nano17, high-bandwidth accelerometers, and an external magnetic tracking system for position measurement [19]. Because the magnetic system is not portable, textures must be brought to the sensing system, and ferromagnetic materials cannot be captured due to interference. Handheld haptic recording systems show great promise for efficient collection of naturalistic surface interaction data, but existing haptic datasets such as the Penn Haptic Texture Toolkit [20] include only a single end-effector and tend to contain only limited visual information about the surfaces.

Recently, haptic sensing technology has been advancing faster than our strategies for interpreting the resulting data. Romano et al. [5] performed classification among a closed set of fifteen surfaces by training an ensemble of one-class ν-SVMs (one for each surface) on features that included normal force, end-effector speed, and frequency-binned vibration power. Yang et al. [21] applied deep learning over similar data to improve upon those results. Strese et al. achieved even better results using features inspired from traditional audio processing such as cepstral analysis [22].

III. DESIGN

The Proton's hardware and software were developed in parallel to cater to our specific needs in acquiring visual and haptic data from surfaces in their natural environments. This section details the design principles and tradeoffs that were considered during this process.

A. Experiment Design

We created the Proton to collect a large-scale dataset. In each data collection episode (an interaction with one particular surface) we gather all relevant information about the surface, making use of all sensors. By collecting data with multiple sensors at almost the same time, we eliminate some variation due to experimental conditions, increasing the generality of the dataset.

After considering prior work, we chose to have a human operator collect the data, rather than a robot. While a robotic platform can make more precise and repeatable motions than a human, it would significantly increase system complexity and cost. In addition, its motor and bearing noise might corrupt the very vibration readings that we are trying to capture with high fidelity [14]. A human guiding the sensing rig ensures that the captured interactions are naturalistic, and including onboard motion tracking allows us to correlate the sensor readings with speed and normal force, as in [19]. Making different motions

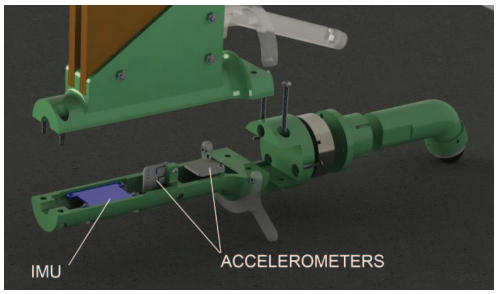

Fig. 2. Cutaway view of the Proton handle's internals.

for each surface complicates data analysis but encourages us to find features that are invariant to such changes.

The user interface to the sensing rig is a web page served by the device itself, typically loaded on a smartphone. The device broadcasts its own wifi network to allow completely portable operation, without external power or communication. With a simple touch interface, the experimenter can start and stop collection of data from any sensor, and view images from the cameras. They can also step through an experimental flow that scripts the activation of the sensors, to facilitate repeatable data collection.

B. Hardware Selection

For the haptic portion of the envisioned data set, we selected sensors that can capture the rich interactions between the sensing rig and real-world surfaces. We have three end-effectors that are mounted one at a time into the Proton. The first end-effector is a SynTouch BioTac, which mimics the multimodal sensing capabilities of the human fingertip [9]. An OptoForce three-axis force sensor, chosen for its small size and ability to directly interact with the surface, is the second end-effector. Lastly, a metal tooling ball without any sensing ability is used as the third end-effector, emulating platforms that have no contact sensors (but may rely on, e.g., accelerometers in a wrist or arm).

To compensate for the different sensing capabilities of the end-effectors, additional sensors are included in the body of the rig. An ATI Mini40 SI-40-2 six-axis force/torque sensor is placed between the end-effector and the handle to provide accurate measurement of contact force. Two ADXL326 high-bandwidth three-axis analog accelerometers are used to capture the high-frequency vibrations of contact. Due to the internal design of this MEMS accelerometer, the z-axis bandwidth (550 Hz) is considerably lower than that of the other two axes (1600 Hz). Therefore we mount two of these accelerometers perpendicular to each other to avoid relying on either sensor's z-axis. The ADXL326 has a sensing range of ± 16 g, which is suitable for measuring vibrations caused by tool-texture interactions [19]. A nine-degree-of-freedom digital inertial measurement unit (IMU), including an L3GD20H three-axis gyroscope and an LSM303 six-axis compass/accelerometer, is used in concert with visual data to reconstruct the motion of the rig. The accelerometer in the IMU differs from the analog ones in that it offers a smaller sensing range (± 4 g), thus lowering noise when reconstructing human motion. Lastly, an ADMP401 analog MEMS microphone records the sounds

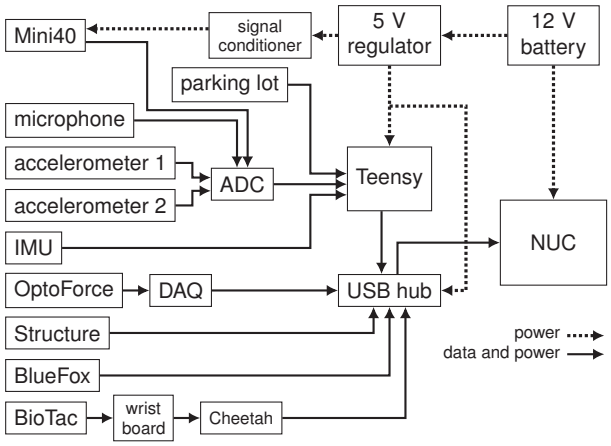

Fig. 3. Wiring diagram of rig and backpack.

of the end-effector contacting the surface. Fig. 2 shows the handle's internal sensors.

A 3D camera, the Structure Sensor [23], helps track the motion of the sensing rig. It requires a minimum distance of 40 cm for depth detection, which dictates the placement of the two cameras (see Section IV-A). A high resolution camera (mvBlueFOX3) captures images and videos of the surface during each data collection episode. It is capable of acquiring images of 1600×1200 resolution at up to 50 Hz. A Computar M2518-MPW2 lens with a focal length of 25 mm provides a relatively large field of view, approximately 10×7.5 cm at a distance of 40 cm.

An Intel Next-Unit-of-Computing (NUC) is used to collect data from the sensors and serve the user interface. This choice provides a battery-powered and small-size computer with a Core i3 CPU, 128 GB SSD, and 4 USB 3.0 ports, suitable for simultaneously interfacing with all of the rig's sensors and recording data in real time. A daughterboard is needed to directly interface with the sensors that do not support USB. For this we use a Teensy 3.1, a USB-based microcontroller that mimics the Arduino architecture [24]. It communicates with the Mini40, analog accelerometers, digital IMU, and microphone, forwarding the data to the NUC for recording.

IV. IMPLEMENTATION

The Proton was built using custom software and hardware to tie together the various off-the-shelf sensing components.

A. Hardware

The Proton consists of a custom-made handheld sensing rig and a backpack containing supporting hardware. All electrical and computational components are located inside the backpack and can be powered by a battery. Fig. 3 shows a schematic.

The design of the handheld portion of the rig was dictated by the requirements of the two cameras: for the BlueFox camera to focus, and for the Structure Sensor to resolve depth measurements, they must be at least 40 cm from the sample. As shown in Fig. 4, the rig is held with two hands, one directing the movement of the end-effector, and the other stabilizing the rig. In the base of the rig, shown in Fig. 2, are the accelerometers,

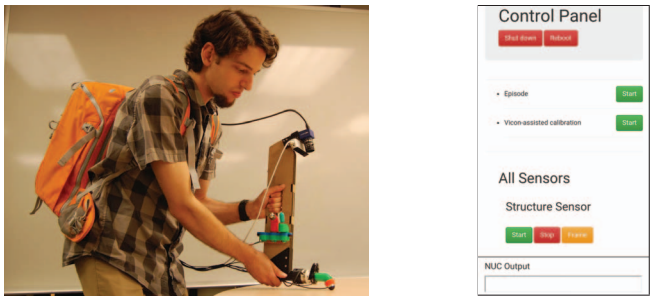

Fig. 4. Left: an experimenter holds the Proton, equipped with the OptoForce end-effector, in contact with a desk surface. Right: the mobile control interface.

IMU, microphone and end-effectors. The Mini40 force/torque sensor is in front, supporting the end-effector socket.

Three finger-shaped end-effectors may be clamped into this socket, differing only in the sensing hardware at the tip. To avoid potential variability in contact force, they are mounted such that the point of surface contact, relative to the body of the rig, is the same. The "parking lot" seen in Fig. 1 provides safe storage of the end-effectors. Each end-effector fits in only one space, so that the NUC can identify end-effectors by their position in the parking lot and automatically label the collected data with the end-effector that was in use.

The BlueFox camera and the Structure Sensor are aimed at the same part of the surface that the end-effector touches, with the end-effector visible in the lower third of the field of view. This arrangement simplifies image registration and depth calibration, although video may also be taken with no end-effector attached to maximize viewing area.

B. Software

We have written a custom software package that runs on the NUC and is in charge of communicating with the sensor hardware, recording data to disk, and interfacing with the experimenter. Designed to be fast, reliable and extensible, the software is written in Rust [25], a new language that emphasizes safety and concurrency without sacrificing speed [26].

This software receives data through different interfaces:

1) BlueFox: Matrix Vision provides the *mvDeviceManager* library. Depending on processor load, we can receive 1600×1200-pixel RGB frames at a rate of 25-30 Hz.

2) Structure: Using the OpenNI 2 library, we record 640×480-pixel grayscale frames at a rate of 30 Hz.

3) OptoForce: We use the *liboptoforce* [27] library, developed at ETH Zurich, to read force data from the OptoForce sensor at a rate of 1000 Hz.

4) BioTac: We ported the BioTac ROS drivers developed by McMahon et al. [28] to our software infrastructure. Data packets are received at 100 Hz.

5) Other sensors: All other sensors are handled by the Teensy, which transmits 3000 packets per second to the NUC. Each packet contains a new reading from the Mini40, microphone and analog accelerometers, plus any available IMU readings (which are cached as they come in).

The software isolates each sensor driver in a separate thread, to increase performance due to parallel processing and to prevent a failure of one device from crashing the program. The software also contains a general framework for scripting interactive experimental paradigms using the Proton's components, which automates setup tasks so that the experimenter need only be concerned with the end-effectors and the surface under study.

V. CALIBRATION

Before conducting any experiments, we designed and executed a calibration procedure to compensate for differences between the actual Proton and its CAD model, and to zero the force/torque sensor.

To allow us to interpret the data recorded from surface interaction, the Proton must be able to track its own motion. In the calibration and validation experiments presented here, an external Vicon motion tracking system was used in place of the onboard IMU and cameras to ensure accuracy and provide a baseline for future implementation of standalone motion tracking.

The goal of the calibration procedure is to unify the various coordinate frames on the sensing rig: the IMU, accelerometers, Mini40, OptoForce, BioTac, and cameras all take measurements that are referenced to their individual coordinate frame's origin and orientation, but they all need reference a common frame for analysis. One way to do this alignment would be open-loop calibration, using our device's CAD model to estimate the translation and rotation offsets. These estimates might be reasonable, but they do not account for small variations in manufacturing and assembly. Furthermore the Vicon coordinate frame (not shown in Figs. 1 or 4) is attached to the sensing rig at an arbitrary location, and we must recover this transformation in order to validate the motion tracking.

For the validation experiment described in the next section, we use only one end-effector (the tooling ball) and no cameras, so the important coordinate frames are the world frame $\mathcal{W}$, the Vicon marker frame $\mathcal{M}$, the body frame $\mathcal{B}$, the force/torque sensor frame $\mathcal{F}$, and the end-effector frame $\mathcal{E}$ (shown in Fig. 5). Notation in this section uses a superscript on vectors and transformation matrices to denote the frame in which they are expressed; e.g., the origin of frame $\mathcal{A}$ expressed in frame $\mathcal{B}$ is $\vec{o}_{\mathcal{A}}^{\mathcal{B}}$. For a vector starting at the origin of frame $\mathcal{A}$ and ending at the origin of $\mathcal{B}$, expressed in a third frame $\mathcal{C}$, a second subscript is used: $\vec{v}_{\mathcal{B}/\mathcal{A}}^{\mathcal{C}}$. R refers to a 3D rotation matrix, while H represents a homogeneous transformation.

As shown in Fig. 5, the body frame is defined with its xy-plane parallel to that of the IMU, its $\hat{z}$-axis pointing up along the Proton handle, and its $\hat{x}$-axis pointing out toward the end-effector. The world frame is stationary, and for now we assume that the surface under investigation is stationary in the world frame, whereas the other four frames are rigidly attached to the rig, and all move together. The Vicon tracking system reports the time-varying position and orientation of the marker frame with respect to the world frame, $H_{\mathcal{M}}^{\mathcal{W}}(t)$. The Mini40 measures forces and torques referenced to its frame $\mathcal{F}$, which is oriented at a known 90-degree angle to the body frame. Therefore, we will transform the Vicon position measurements to find the pose of the body frame. We use two calibration procedures to recover different parameters: one with the end-effector fixed, and another with the end-effector free.

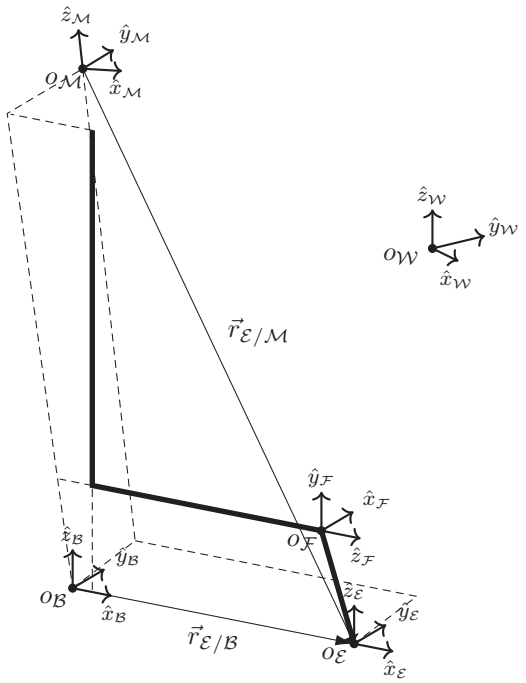

Fig. 5. Coordinate frames. The thick line shows an outline of the Proton (not to scale). The world frame $\mathcal{W}$ is stationary, and the other four frames are rigidly attached to the rig. The body frame $\mathcal{B}$ and the end-effector frame $\mathcal{E}$ are aligned and have colinear x-axes. The origin of $\mathcal{B}$ is in the yz-plane of the Vicon marker frame $\mathcal{M}$.

Parameter	Symbol	Value
Vector from $\mathcal{M}$ to $\mathcal{E}$	$\vec{r}^{\,\mathcal{M}}_{\mathcal{E}/\mathcal{M}}$	$[252.2, 30.34, -545.1]$ mm
Rotation from $\mathcal{M}$ to $\mathcal{B}$	$R^{\mathcal{B}}_{\mathcal{M}}$	$\begin{bmatrix} 0.991 & -0.024 & -0.130 \\ 0.017 & 0.998 & -0.058 \\ 0.132 & 0.055 & 0.990 \end{bmatrix}$
Mini40 force bias	$\vec{b}^{\,\mathcal{F}}_{F}$	$[\ 1.552, -4.345, -0.182]$ N
Mini40 torque bias	$\vec{b}^{\,\mathcal{F}}_{\tau}$	$[-0.106,\ \ 0.037, -0.029]$ Nm
Tooling ball end-effector mass	m_t	150.3 g
Tooling ball end-effector CoM	$\bar{r}^{\,\mathcal{F}}_{t}$	$[-2.29, -3.01, 34.8]$ mm
OptoForce end-effector mass	m_o	110.2 g
OptoForce end-effector CoM	$\bar{r}^{\,\mathcal{F}}_{o}$	$[-5.56, 1.41, 21.8]$ mm
BioTac end-effector mass	m_b	113.8 g
BioTac end-effector CoM	$\bar{r}^{\,\mathcal{F}}_{b}$	$[-3.66, 1.86, 21.0]$ mm

TABLE I
CALIBRATION RESULTS.

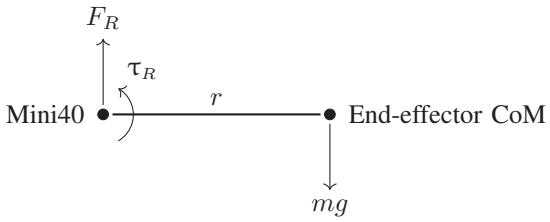

Fig. 6. Free-body diagram of the force/torque sensor and end-effector during free end-effector calibration. The Mini40 measures the reactions $\vec{F}_R$ and $\vec{\tau}_R$. Sensor bias is not shown.

A. Fixed End-effector Calibration

The first calibration procedure determines the vector from the marker frame to the end-effector frame, $\vec{r}^{\,\mathcal{M}}_{\mathcal{E}/\mathcal{M}}$. We fix the tooling ball end-effector in place using a socket and rotate the Proton in all directions while recording $H^{\mathcal{W}}_{\mathcal{M}}(t)$. Each world-frame position measurement will be at a constant distance from the end-effector, lying on a sphere whose radius is the distance from the origin of $\mathcal{M}$ to the origin of $\mathcal{E}$:

$$\vec{o}^{\,\mathcal{W}}_{\mathcal{M}}(t) = \vec{o}^{\,\mathcal{W}}_{\mathcal{E}} + \vec{r}^{\,\mathcal{W}}_{\mathcal{M}/\mathcal{E}}(t) \qquad \forall t \tag{1}$$

Furthermore, no matter the orientation of the Proton, the vector we seek remains constant with respect to the moving frames. We fit a sphere to the time-varying $\vec{o}^{\,\mathcal{W}}_{\mathcal{M}}$ measurements, using least-squares minimization and RANSAC to exclude outliers. This fit recovers both the center of the sphere, $\vec{o}^{\,\mathcal{W}}_{\mathcal{E}}$, and its radius, $||\vec{r}_{\mathcal{E}/\mathcal{M}}||$, with 68% inliers and RMS error of 8.4 mm. From there we can recover the vector:

$$\vec{r}^{\,\mathcal{M}}_{\mathcal{E}/\mathcal{M}} = H^{\mathcal{M}}_{\mathcal{W}}(t)\ \vec{r}^{\,\mathcal{W}}_{\mathcal{E}/\mathcal{M}}(t) \qquad \forall t \tag{2}$$

We average the results of $H^{\mathcal{M}}_{\mathcal{W}}(t)\ \vec{r}^{\,\mathcal{W}}_{\mathcal{E}/\mathcal{M}}(t)$ over the calibration period to estimate $\vec{r}^{\,\mathcal{M}}_{\mathcal{E}/\mathcal{M}}$, as given in Table I.

B. Free End-effector Calibration

The Mini40 sensor readings are always offset by a constant bias, independent of the applied force and torque. The second calibration recovers this bias, the orientation of the body frame with respect to the marker frame, and the mass and center of mass of each end-effector (which are used to perform gravity compensation). We record position, orientation, force, and torque while slowly rotating the Proton in all directions in free space – that is, without the end-effector contacting anything. The procedure was repeated twice for each end-effector, so fit values and errors in this section are aggregated over the six runs. Under these conditions, the expected force and torque measurements can be described by these simple equations (see the free-body diagram in Fig. 6):

$$\vec{F}^{\,\mathcal{B}}_{R}(t) = \vec{b}^{\,\mathcal{B}}_{F} + m\vec{g}^{\,\mathcal{B}}(t) \tag{3}$$

$$\vec{\tau}^{\,\mathcal{B}}_{R}(t) = \vec{b}^{\,\mathcal{B}}_{\tau} + \vec{r}^{\,\mathcal{B}} \times m\vec{g}^{\,\mathcal{B}}(t) \tag{4}$$

The total force on the sensor (3) comes from a constant offset $\vec{b}_F$, the sensor's internal force bias, plus the force of gravity, $m\vec{g}(t)$; here, m is the mass of the end-effector, and $\vec{g}(t)$ is the constant-magnitude gravitational acceleration vector, whose direction varies over time. Similarly, the total torque (4) comes from the internal torque bias $\vec{b}_\tau$ plus the torque contributed by gravity acting on the mass of the end-effector, $\vec{r}\times m\vec{g}(t)$, where $\vec{r}$ is the constant vector from the sensor to the end-effector's center of mass.

(3) has the form of a sphere equation, so fitting a sphere to the force data gives the solution for $\vec{b}_F$ as the center and m as the radius. The fit values, which have an average of 93.6% inliers ($\pm$8.5 percentage points std) and RMS error of 0.079 N ($\pm$0.013 N std), are shown in Table I. Then we compare gravity vectors to determine the relative orientation of the two frames. In the body frame, the direction of gravity is given by the force measurement minus the force bias. In the world frame, we know that gravity points along the negative $\hat{z}$-axis, so we transform this vector into the marker frame using the Vicon

measurements. We solve for the rotation between those two sets of gravity vectors using the SVD method of [29]. This result gives the desired transformation between $\mathcal{M}$ and $\mathcal{B}$.

To extract the torque bias, we substitute (3) into (4):

$$\vec{\tau}_R(t) = \vec{b}_\tau + \vec{r} \times (\vec{F}_R(t) - \vec{b}_F) \tag{5}$$

Then we apply robust least squares estimation to solve (5) for $\vec{b}_\tau$ and $\vec{r}$. The fit values, which have an average σ of 0.00095 Nm ($\pm$0.00036 Nm std), are shown in Table I.

C. Synthesis

After performing these two calibrations, we use the results to construct two homogeneous transformations: $H_{\mathcal{B}}^{\mathcal{M}}$, which gives the pose of the body frame in the marker frame, and $H_{\mathcal{E}}^{\mathcal{B}}$, which gives the pose of the end-effector frame in the body frame.

We have free choice of the origin of $\mathcal{B}$. To reduce the number of unknowns, we specify that the origin of $\mathcal{E}$ is on the x-axis of $\mathcal{B}$ with no rotation between those two frames, and that the origin of $\mathcal{B}$ is in the yz-plane of $\mathcal{M}$ (refer back to Fig. 5). Then the combined transformation $H_{\mathcal{B}}^{\mathcal{M}} H_{\mathcal{E}}^{\mathcal{B}}$ has $\vec{r}_{\mathcal{E}/\mathcal{M}}^{\mathcal{M}}$ as its translational component, leading to the constraint:

$$\begin{bmatrix} & R_{\mathcal{B}}^{\mathcal{M}} & & 0 \\ & & & y \\ & & & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & I_3 & & x \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \vec{r}_{\mathcal{E}/\mathcal{M}}^{\mathcal{M}} \\ 1 \end{bmatrix} \tag{6}$$

Solving this equation symbolically yields the unknown matrix elements x, y, and z.

Table I shows a summary of the calibration results. The top section shows the results from the fixed calibration, and the lower two sections show results from the free calibration. As expected, all three end-effectors have similar masses, but the tooling ball is the heaviest and its center-of-mass is farthest from the force/torque sensor. These values are reasonable but do not match the CAD model exactly due to fabrication variations.

VI. EXPERIMENT

After completing the calibration process, we conducted a proof-of-concept surface discrimination experiment to confirm the quality of the data collected using the Proton. This section details the experimental design and results.

We used the rig with the tooling ball end-effector and the Vicon system to gather contact force, vibration and end-effector pose data from the following five surfaces: ABS plastic, paper plate, folder (thin cardboard), medium-density fiberboard, and canvas. The surface samples are shown in Fig. 7 and come from the dataset in [20]. We purposely chose some surfaces that feel distinct and others that feel similar to one another. Here, the Vicon tracker is used instead of the onboard IMU in order to provide the maximum accuracy for validating the haptic sensors, and to generate ground-truth data for future work on self-contained motion tracking.

During data collection, the Proton Pack end-effector was dragged in straight lines back and forth over the surface, varying the speed and applied normal force. There were five trials for each surface, each ten seconds long. The surface was placed on a horizontal table, so it is known to be parallel to the world xy-plane. Since the Vicon tracker's output stream is not synchronized with the other sensor streams, each trial started and ended with two taps. These taps can be clearly identified in both the force and position data, so the streams can be manually aligned in time.

Fig. 7. Surface samples used in the validation experiment.

The data were processed in several ways before the handoff to machine learning. First, the sensor streams were synchronized (this includes upsampling the Vicon data to match the 3000 Hz sample rate of the Mini40 sensor and accelerometers) and automatically trimmed, using the taps to mark the beginning and end of the streams. Next, the pose measurements were transformed to give the pose of the body frame:

$$H_{\mathcal{B}}^{\mathcal{W}}(t) = H_{\mathcal{M}}^{\mathcal{W}}(t)\ H_{\mathcal{B}}^{\mathcal{M}} \tag{7}$$

The force and acceleration readings, measured with respect to the body frame, were rotated to reference the world frame.

$$\vec{F}^{\,\mathcal{W}}(t) = R_{\mathcal{B}}^{\mathcal{W}}(t)\ \vec{F}^{\,\mathcal{B}}(t) \tag{8}$$

$$\vec{a}^{\,\mathcal{W}}(t) = R_{\mathcal{B}}^{\mathcal{W}}(t)\ \vec{a}^{\,\mathcal{B}}(t) \tag{9}$$

Once in the world frame, where the direction of gravity is known, we can perform gravity compensation on the force measurements to remove the contribution from the end-effector's own weight.

$$\vec{F}_c^{\,\mathcal{W}}(t) = \vec{F}^{\,\mathcal{W}}(t) + m\vec{g}^{\,\mathcal{W}} \tag{10}$$

$\vec{F}_c^{\,\mathcal{W}}(t)$ can be decomposed into normal force (the vertical component, since each surface was placed on a level table), and tangential force (the horizontal component, projected onto the the opposite of the end-effector velocity to isolate friction).

Each example for training or testing the machine learning algorithm was a 50 ms window. To avoid training the classifier on periods of time where the operator was changing direction, or while the end-effector was not in constant contact with the surface, these windows were selected from periods where the end-effector was moving with at least 20 mm/s of speed and contacting the surface with at least 3 N of normal force. From 250 s of data, 103.75 s met these thresholds, which yielded 2075 time windows. These were randomly partitioned into a training set of 1689 examples and a test set of 386 examples.

The learning was done with a multiclass ν-SVM using a radial basis function (RBF) kernel, parametrized by ν (allowed

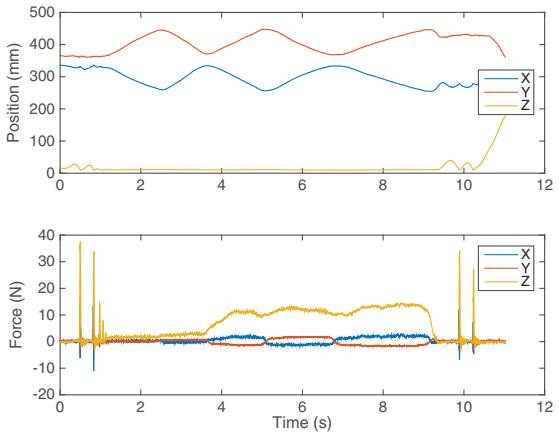

Fig. 8. Typical position (top) and force (bottom) data generated by dragging the Proton's tooling ball end-effector across a folder texture sample.

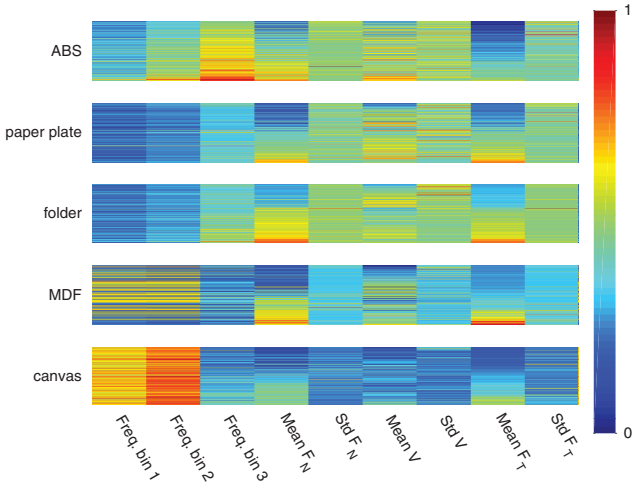

Fig. 9. Feature vectors in the training set. Each row is one example, with 9 feature columns.

misclassifications) and γ (inverse variance of the kernel). For each example we calculated binned acceleration frequencies, following the "perceptual binning" scheme of [5], and summary statistics (mean and standard deviation) of the normal force, F_N, tangential force, F_T, and speed, v. This feature extraction was parametrized by n (number of frequency bins) and α (width of the frequency bins, described in [5]). To choose optimal values for ν, γ, n, and α, we performed grid search with five-fold cross-validation.

Features are normalized by subtracting the mean and dividing by the range (calculated from the training set). Fig. 9 shows a visualization of the normalized features in the training set, with one row per example, sorted by an average of the normal force, tangential force, and end-effector speed. The first three features are the frequency-binned vibration power, followed by mean and standard deviation of normal force, speed, and tangential force respectively. It appears that each surface has a distinctive vibration fingerprint, in terms of average amplitude as well as dependence on force and speed.

The grid search and cross-validation revealed the best choices of the parameters, shown in Table II, which resulted in 85.2% average accuracy on the validation set and 82.5% accuracy on the test set. Numerical measures of the model performance on all five materials, slightly inflated due to treating the five-class SVM as a one-vs-all classifier, are shown in Fig. 10 and Table III. Performance is significantly higher than chance (20% accuracy), although we see some confusion among surfaces, especially the paper plate and folder. Compared to [5] (72.4% accuracy among fifteen surfaces) and [22] (95% accuracy among 69 textures), our results are less impressive – we achieve higher absolute accuracy than [5], but with a smaller set of surfaces. We expect to achieve better accuracy after improving data quality and feature engineering. In particular, we believe the large tooling ball evoked less interesting signals than the smaller tool tips used in these previous studies.

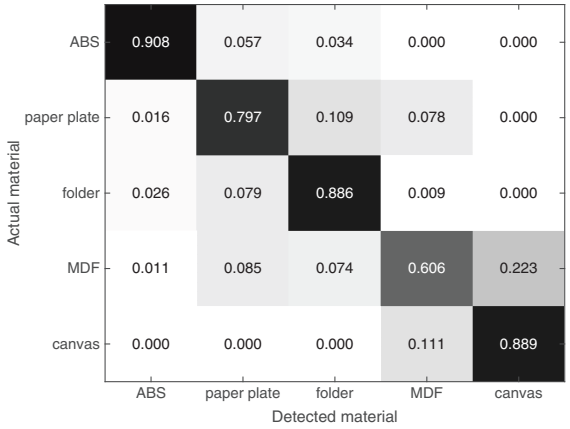

Fig. 10. Confusion matrix. Entries are normalized by the total number of samples of each material.

Parameter	Minimum	Maximum	Increment	Best value
ν	0.05	0.4	0.05	0.25
γ	1	20	2	7
n	1	10	2	3
α	0.1	0.5	0.05	0.15

TABLE II

BEST LEARNING PARAMETERS, AS DECIDED BY CROSS-VALIDATION.

Surface	Precision	Recall	F_1 score
ABS	0.940	0.908	0.924
paper plate	0.699	0.797	0.745
folder	0.856	0.886	0.871
MDF	0.740	0.606	0.667
canvas	0.842	0.889	0.865

TABLE III

TEST SET PERFORMANCE OF THE BEST MODEL FROM CROSS VALIDATION.

VII. CONCLUSIONS

This paper presented the design and implementation of the Proton, a versatile visuo-haptic sensing system that will be used to gather a dataset we hope will be broadly useful to robotics researchers and their robotic platforms. The Proton

contains sensors that span relevant parts of the haptic and visual perceptual space, designed to gather the types of data encountered by many different robots, which may have haptic sensors, visual sensors (2D or 3D), or both. Some robots may query our database in order to identify an unknown surface. Others may be trying to predict haptic sensations given visual data, or the reverse.

There is still some work to do before we are ready to collect and publish a full dataset. We need to integrate the BioTac sensor and address reliability issues in reading packets from the Teensy. On the software side, we will develop a data analysis pipeline to validate the data that we are collecting.

Once these tasks are completed, the main work will be in two areas: data collection and machine learning.

1) Data collection: First we will need to survey a wide range of surfaces with our sensing rig. We plan to publish the dataset, which will include surfaces from the existing Yale-CMU-Berkeley (YCB) Object and Model set [30], as open source for other researchers to use as well.

2) Machine learning: As soon as more data is available to analyze, our goals will be supervised/unsupervised surface categorization, regression to recover human ratings from sensor data, and finding cross-modality correspondences between the visual data and the haptic sensors. Because such a correlative dataset has not been published before, one key contribution will be methods for drawing cross-modal correspondences between the visual appearance of a texture and its haptic feel.

Another important contribution is the dataset itself, which can be federated with other publicly available visual and haptic datasets, as well as used directly by robots with various sensing capabilities to identify unfamiliar surfaces. We look forward to opening the Proton dataset for contributions, federation, and research by the haptics, vision, and mobile robotics communities.

ACKNOWLEDGMENTS

The authors thank Naomi Fitter and Jennifer Hui for their design advice and consultation on the use of the SynTouch BioTac, Eric Young for his role in data collection, and Abhinav Rajvanshi and Sarah Allen for editing assistance.

REFERENCES

[1] Defense Advanced Research Projects Agency. DARPA Robotics Challenge Home. http://www.theroboticschallenge.org/, 2013.

[2] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. Dedonato, R. Du, S. Feng, P. Franklin, M. Gennert, J. P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, T. Padir, F. Polido, G. G. Tighe, and X Xinjilefu. No falls, no resets: Reliable humanoid behavior in the DARPA Robotics Challenge. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, pages 623–630, 2015.

[3] S. A. Stansfield. Primitives, features, and exploratory procedures: Building a robot tactile perception system. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 1274–1279, 1986.

[4] Susan J. Lederman and Roberta L. Klatzky. Hand movements: a window into haptic object recognition. *Cognitive Psych.*, 19(3):342–368, 1987.

[5] Joseph M. Romano and Katherine J. Kuchenbecker. Methods for robotic tool-mediated haptic surface recognition. In *Proc. IEEE Haptics Symp.*, pages 49–56, 2014.

[6] Jivko Sinapov, Vladimir Sukhoy, Ritika Sahai, and Alexander Stoytchev. Vibrotactile recognition and categorization of surfaces by a humanoid robot. *IEEE Trans. on Robotics*, 27(3):488–497, 2011.

[7] Yuka Mukaibo, Hirokazu Shirado, Masashi Konyo, and Takashi Maeno. Development of a texture sensor emulating the tissue structure and perceptual mechanism of human fingers. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2565–2570, 2005.

[8] Nawid Jamali and Claude Sammut. Majority voting: material classification by tactile sensing using surface texture. *IEEE Transactions on Robotics*, 27(3):508–521, 2011.

[9] Jeremy A. Fishel and Gerald E. Loeb. Bayesian exploration for intelligent identification of textures. *Frontiers in Neurorobotics*, 6, 2012.

[10] Vivian Chu, Ian McMahon, Lorenzo Riano, Craig G. McDonald, Qin He, Jorge Martinez Perez-Tejada, Michael Arrigo, Trevor Darrell, and Katherine J. Kuchenbecker. Robotic learning of haptic adjectives through physical interaction. *Robotics and Auton. Syst.*, 63(3):279–292, Jan. 2015.

[11] Roberta L. Klatzky, Susan J. Lederman, Cheryl Hamilton, and Molly Grindley. Perceiving surface roughness through a probe: effects of applied force and probe diameter. In *Proc. ASME Dynamic Systems and Control Division*, volume 2, 2000.

[12] Danfei Xu, Gerald E. Loeb, and Jeremy A. Fishel. Tactile identification of objects using Bayesian exploration. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3056–3061, 2013.

[13] Nathan F. Lepora, Uriel Martinez-Hernandez, and Tony J. Prescott. Active touch for robust perception under position uncertainty. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3020–3025, 2013.

[14] William McMahan and Katherine J. Kuchenbecker. Spectral subtraction of robot motion noise for improved vibrotactile event detection. In P. Isakoski and J. Springare, editors, *Haptics: Perception, Devices, Mobility, and Communication: Proc. EuroHaptics, Part I, volume 7282 of Lecture Notes in Computer Science*, pages 326–337, 2012.

[15] Dinesh K. Pai and Peter Rizun. The WHaT: a wireless haptic texture sensor. In *Proc. Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 3–9, 2003.

[16] Sheldon Andrews and Jochen Lang. Interactive scanning of haptic textures and surface compliance. In *Proc. IEEE Int. Conf. on 3-D Digital Imaging and Modeling*, pages 99–106, 2007.

[17] Edoardo Battaglia, Giorgio Grioli, Manuel G. Catalano, Marco Santello, and Antonio Bicchi. ThimbleSense: an individual-digit wearable tactile sensor for experimental grasp studies. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2728–2735, 2014.

[18] Raphael Höver, Matthias Harders, and Gábor Székely. Data-driven haptic rendering of visco-elastic effects. In *Proc. IEEE Symp. on Haptic Interfaces for Virtual Env. and Teleoperator Syst.*, pages 201–208, 2008.

[19] Heather Culbertson, Juliette Unwin, Benjamin E. Goodman, and Katherine J. Kuchenbecker. Generating haptic texture models from unconstrained tool-surface interactions. In *Proc. IEEE World Haptics Conf.*, pages 295–300, 2013.

[20] Heather Culbertson, Juan José López Delgado, and Katherine J. Kuchenbecker. One hundred data-driven haptic texture models and open-source methods for rendering on 3D objects. In *Proc. IEEE Haptics Symp.*, pages 319–325, February 2014.

[21] Yang Gao, Lisa Anne Hendricks, Katherine J. Kuchenbecker, and Trevor Darrell. Deep Learning for Tactile Understanding From Visual and Haptic Data. Accepted to *IEEE Int. Conf. on Robotics and Automation*, 2016.

[22] Matti Strese, Clemens Schuwerk, and Eckehard Steinbach. Surface classification using acceleration signals recorded during human freehand movement. In *Proc. IEEE World Haptics Conf.*, pages 214–219, 2015.

[23] The Structure sensor. http://structure.io.

[24] Teensy 3.1. https://www.pjrc.com/teensy/teensy31.html.

[25] The Rust programming language. https://www.rust-lang.org.

[26] Alexis Beingessner. You can't spell trust without Rust. Master's thesis, Carleton University, 2015.

[27] Ralf Kaestner and Mark Hoepflinger. liboptoforce. https://github.com/ethz-asl/liboptoforce.

[28] Vivian Chu, Lorenzo Riano, Ian McMahon, Karen Qin, and Craig McDonald. IanTheEngineer/Penn-haptics-bolt: Penn Haptics team's repository for BOLT related work. URL: *https://github.com/IanTheEngineer/Penn-haptics-bolt*. Accessed May 2016.

[29] Paul J. Besl and Neil D. McKay. Method for registration of 3-D shapes. In *Proc. SPIE 1611*, pages 586–606. International Society for Optics and Photonics, 1992.

[30] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols. *Robotics and Automation Magazine (RAM), Special Issue on Replicable and Measurable Robotics Research*, 22(3):36–52, Sept 2015.