December 2019

TACTILE TEXTURES FOR BACK OF SCREEN GESTURE DETECTION USING MOTION SENSOR DATA AND MACHINE LEARNING

Michael Xuelin Huang

Scott Jensen

Shumin Zhai

TACTILE TEXTURES FOR BACK OF SCREEN GESTURE DETECTION USING MOTION SENSOR DATA AND MACHINE LEARNING

ABSTRACT

A computing device is described that uses motion data from motion sensors to detect gestures or user inputs, such as out-of-screen user inputs for mobile devices. In other words, the computing device detects gestures or user touch inputs at locations of the device that do not include a touch screen, such as anywhere on the surface of the housing or the case of the device. A tactile texture is applied to a housing of the computing device or a case that is coupled to the housing. The tactile texture causes the computing device to move in response to a user input applied to the tactile texture, such as when a user's finger slides over the tactile texture. A motion sensor (e.g., an inertial measurement unit (IMU), accelerometer, gyroscope, etc.) generates motion data in response to detecting the motion of the computing device. The motion data is processed by an artificial neural network to infer attributes of the user input. In other words, the computing device applies a machine-learned model to the motion data (also referred to as sensor data or motion sensor data) to classify or label the various attributes, characteristics, or qualities of the input. In this way, the computing device utilizes machine learning and motion data to classify attributes of the user input or gesture utilizing motion sensors without the need for additional hardware, such as touch-sensitive devices and sensors.

DESCRIPTION

Techniques are described that enable a computing device to detect gestures performed on a surface of a computing device that is not touch-sensitive, such as the back of the computing device, using data from one or more motion sensors. Examples of computing devices include, but are not limited to, smartphones, tablets, watches, fitness trackers (e.g., wrist-worn, ankle-

worn, waist-worn, or other devices worn by users that track various movement or other athletic activity), counter-top computing devices, laptops, or any other computing device.

The computing device includes a touch-sensitive device (e.g., a touchscreen) on the front, one or more processors, and one or more motion sensors. Examples of processors include, but are not limited to, digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry.

According to techniques of this disclosure, the computing device may include a tactile texture on one or more surfaces that are not touch-sensitive. For example, the front of the computing device may include a touch-sensitive device (e.g., a touchscreen or touchpad) while the back of computing device includes a tactile texture but does not include a touch-sensitive device. In some examples, the tactile surface may be applied to a back surface of the housing of the computing device and/or a surface of a case that is physically coupled to the back of the housing.

FIG. 1 illustrates one example of a tactile texture. In the example of FIG. 1, tactile texture 2 includes a force directed wave that includes a plurality of protrusions 4. Different sides of each of protrusions 4 may have different shapes or geometries. For example, as illustrated in FIG. 1, one side of protrusions 4 may be an arc of a circle (e.g., rounded) while the other side of protrusions 4 may be an incline (e.g., having a constant slope). In some examples, the radius of the circle may be approximately 1 mm and the length of the incline may be approximately 4 mm. In some examples, a steeper slope of the incline may increase resistance and hence force required to perform a user input. It should be understood that the sides of protrusions 4 may have different geometries and/or dimensions than the geometries and dimensions illustrated in FIG. 1. For example, protrusions 4 may include an incline having a first slope on one side and another incline having a different slope on the other side; a series of bumps on one side and a relatively

smooth surface on the other side, among many other geometries. While illustrated as having different geometries in the Y-direction, in some examples, protrusions 4 may have different geometries in the X-direction to detect left and right movements (additionally or alternatively to detecting upwards and downwards movements).

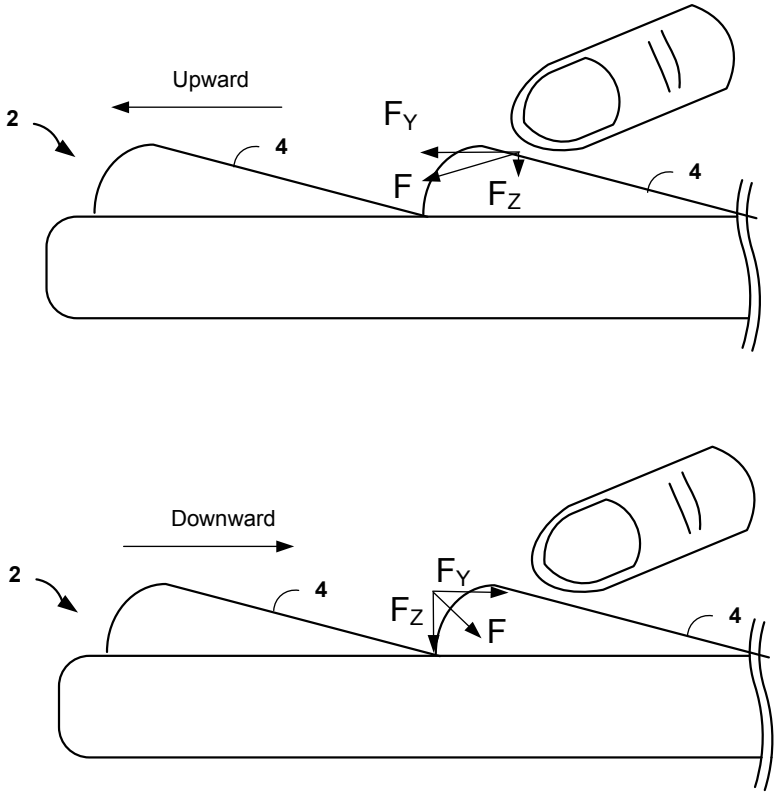

FIG. 1

A user of the computing device may perform a gesture or user input across tactile texture 2. As one example, the user may perform a swipe, pinch, tap, double-tap, or any other type of gesture across tactile texture 2 (e.g., using his/her finger, a stylus, or other input mechanism that makes physical contact with the computing device). In the example of FIG. 1, an upward gesture may create an overall force F in line with the angle of the input device (e.g., finger), while a downward gesture may create an overall force F that points towards the center of the circle. In some examples, the Y and Z components (F_Y and F_Z, respectively) of force F may be different depending on the direction of the gesture (e.g., upward vs downward). For example, the normal

force F_Z that is normal to the housing or case of the computing device for an upward gesture may be different that the normal force F_Z for a downward gesture. In some examples, utilizing a rounded geometry on one side of protrusions 4 and a constant incline geometry on the other side of protrusions 4 may maximize the difference in the normal force F_Z, which may enable the computing device to more accurately differentiate between gestures.

FIG. 2 illustrates another example of a tactile texture. In the example of FIG. 2, tactile texture 12 includes a plurality of rhythmic stripes 14A and 14B (collectively, rhythmic stripes 14). While tactile texture 12 is shown with two different rhythmic stripes, tactile texture 12 may include any number of rhythmic stripes.

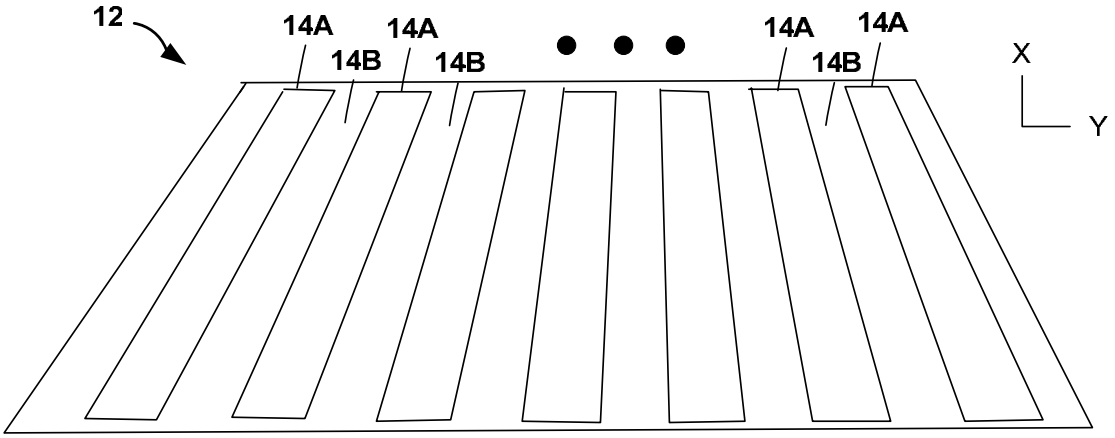

FIG. 2

Rhythmic stripes 14 may be manufactured via any suitable process. In one example, tactile texture 12 includes a smooth, flat acrylic sheet. In such examples, rhythmic stripes 14A may be formed by etching stripes in the acrylic sheet.

In some examples, rhythmic stripes 14A and 14B have a different degrees of roughness. That is, the coefficient of friction of rhythmic stripe 14A may be different than the coefficient of friction of rhythmic stripes 14B. In some examples, the coefficient of friction of rhythmic stripes 14B is higher than the coefficient of friction of rhythmic stripes 14A. While FIG. 2 illustrates tactile texture 12 as including rhythmic stripes of alternating coefficients of friction in the Y-

direction, in some examples, tactile texture 12 may additionally or alternatively include rhythmic stripes having different coefficients of friction in the X-direction.

Rhythmic stripes 14A and 14B may have the same width or different widths. The width of one rhythmic stripe 14A may be the same or different than the width of another rhythmic stripe 14A. Similarly, the width of one rhythmic stripe 14B may be the same or different than the width of another rhythmic stripe 14B. That is, the interval of the rhythmic stripes 14 may be spread evenly or unevenly. In one example, the width of rhythmic stripes 14 may be a gradient, such that rhythmic stripes 14 may be one width at one end of the computing device and change in width over the length of the computing device (e.g., starting narrow and getting wider as the rhythmic stripes go from top to bottom, or vice versa).

A user may perform a user input over tactile texture 2 or 12 and the computing device may move in response to the forces that are exerted on tactile texture 2 or 12 by the user input. In some examples, the computing device includes one or more motion sensors configured to detect motion of the computing device. Examples of motion sensors include an accelerometer, a gyroscope, an inertial measurement unit (IMU), a magnetometer, among others. In one example, as the user moves his or her finger over protrusions 4 of tactile texture 2, a motion sensor of the computing device may detect the motion caused by the normal force F_Z. In another example, a motion sensor may detect motion of the computing device traverses rhythmic stripes 14 of tactile texture 12. For example, because rhythmic stripes 14A and 14B may have different coefficients of friction, the force exerted on the computing device and the resulting motion of the computing device may be different as the user's finger traverses the different rhythmic stripes 14. That is, the user input may produce alternating frictional forces as the user input traverses over rhythmic stripes 14 having different coefficients of friction.

In some examples, an IMU (e.g., including a 3-axis accelerometer and a 3-axis gyroscope) outputs motion data with six values (e.g., an acceleration value for each of the three

axes and a gyroscope value for each of the three axes) each time the IMU samples the motion of the computing device. The computing device may store the motion data as a one-dimensional vector (or as any other type of data structure) with six sensor values or elements for each time the IMU samples the motion. The IMU may sample the motion every 1 millisecond, every 2 milliseconds, every 5 milliseconds, and so on. In some examples, the vector includes data for multiple samples (e.g., 10, 20, 50, 100 samples). In this way, the vector may include six sensor values for each of the plurality of samples.

In some examples, the computing device applies a machine-learned model to the motion data to determine the attributes or characteristics of the user input. The model may be trained via supervised or unsupervised learning. In some examples, the model includes a neural network, such as a convolutional neural network (CNN). Additional examples of machine learning algorithms include clustering algorithms, decision-tree algorithms, regression algorithms, as only a few examples. In one example, the computing device may apply the vector of motion data to the CNN. In other words, the input to the CNN includes, in some examples, a one-dimensional vector of motion data over time (e.g., the vector may include six values for each time the IMU is sampled within a period of time).

The computing device may apply a multi-layer CNN to the input vector. For example, the CNN may include 2, 3, 4, or more layers. The layers may be partially or fully connected. For each convolution layer, the computing device applies a filter to the input for that layer. As one example, the computing device may multiply the one-dimensional input vector (also referred to as a matrix) by a convolution filter and output a convoluted vector (or matrix) for the first convolution layer. The computing device may input the convoluted vector output by the first layer as an input to the second layer, apply another filter, and output a new convoluted vector, and so on for each convolution layer of the CNN.

The computing device may determine a plurality of attributes of the user input based on the CNN. In other words, in one example, the CNN receives the one-dimensional vector (e.g., with six values for each sample of the IMU) as the input and outputs a plurality of values representative of the attributes of the user input.

In some examples, because different attributes of the user input may affect the motion data simultaneously, the computing device may apply the multi-layer CNN to jointly learn the mappings from motion data onto these interrelated attributes in a multi-task learning paradigm. In one example, the computing device determines the different attributes of the user input by utilizing one or more shared convolutional layers, which may enable the computing device to detect or extract common patterns that are indicative across user input attributes. Following the shared layers, in some examples, the CNN includes individual layers for each attribute of the user input to extract the attribute-dependent patterns.

In some examples, the CNN outputs data indicating one or more attributes or characteristics of the user input, such as a direction of the user input, a length or distance of the user input, and/or a location of the user input. For example, the CNN may output data to the computing device, which may classify the direction of the gesture or user input as one of upward, downward, or none.

As another example, the CNN may output data indicating a number of protrusions 4 or rhythmic stripes 14 traversed by the user input, which may enable the computing device to determine the total distance of the user input. For example, when the computing device includes tactile texture 12 having rhythmic stripes 14, the CNN may analyze the Y-axis of the accelerometer signal to detect the peak and valley of the acceleration. The CNN may output a count of the peak-valley pairs of the Y-axis of the accelerometer signal, such that the computing device may multiply the count of the peak-valley pairs and multiply the count by the width of rhythmic stripes 14 to determine the distance of the user input.

The CNN may, in some examples, output data indicating a location of the user input. For example, when the computing device includes tactile texture 12 with rhythmic stripes 14, the interval of rhythmic stripes may not be spread evenly. In such examples, the CNN may be tuned or trained based on the IMU signals and the specific pattern and widths of rhythmic stripes 14. In this way, the CNN may output data indicating a specific location of the user input.

The computing device may perform one or more actions based on the output of the CNN. The computing device may scroll (e.g., up, down, left, right) through a graphical user interface in response to detecting an upward or downward gesture. As another example, the computing device may adjust a slider based on the gesture. The on-screen scrolling distance or slider distance may correspond to the distance of the user input, which may provide the user with a sense of direct manipulation of the computing device.

The techniques of this disclosure may enable a user of the computing device to perform one-handed interactions with the computing device (e.g., interacting with the computing device with the same hand that is holding the device). For example, the user may only have one hand available (e.g., while holding a bag, child, or other object). In another example, the techniques of this disclosure may enable the user to more easily interact with the computing device while lying down by using most of his/her fingers to grasp the phone and using one finger (e.g., the index finger) to interact with the computing device. Utilizing tactile textures may also provide a haptic feedback to users, which may enable the user to perform relatively precise interactions without relying on visual feedback.

In some scenarios, the techniques of this disclosure may enable the device to detect gestures (e.g., swiping) across the back surface or other surfaces of the computing device with little to no additional hardware. For example, by applying a machine-learned model (such as a CNN model) to the motion data, the computing device may determine one or more attributes of a touch input using existing motion sensors without utilizing a touch-sensitive device, such as a

touchscreen or touchpad. Detecting user inputs and determining the attributes of the input without utilizing a touch-sensitive device may reduce the cost of the computing device and/or reduce the power consumed by the device, for example, by detecting user inputs without activating a touch-sensitive device.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in any combination of the following documents:

1. Le et al., *InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones*, User Interface Software and Technology Symposium 2018 (UIST '18), October 14-17 2018, DOI 10.1145/3242587.3242605
2. De Luca et al., *Back-of-device Authentication on Smartphones*, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13), April 27-May 2, 2013, DOI: 10.1145/2470654.2481330.
3. Le et al., *Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage*, Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI'16), October 23-27, 2016, DOI 10.1145/2971485.2971562.
4. Corsten et al., *BackXPress: Using Back-of-Device Finger Pressure to Augment Touchscreen Input on Smartphones*. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17), May 6-11, 2017, DOI 10.1145/3025453.3025565.
5. Holman et al., *Unifone: Designing for Auxiliary Finger Input in One-handed Mobile Interactions,* Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13), February 10-13, 2013, DOI 10.1145/2460625.2460653
6. Wong et al., *Back-Mirror: Back-of-Device One-Handed Interaction on Smartphones,* Proceedings of the SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications Article No. 10 (SA'16), December 5-8, 2016 DOI 10.1145/2999508.2999522.

7. Sun et al., *VSkin: Sensing Touch Gestures on Surfaces of Mobile Devices Using Acoustic Signals,* Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18), October 29 - November 02, 2018, DOI 10.1145/3241539.3241568.
8. Tung et al., *Expansion of Human–Phone Interface By Sensing Structure-Borne Sound Propagation,* Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'16), June 26-30, 2016 DOI 10.1145/2906388.2906394.
9. Reyes et al., *SynchroWatch: One-Handed Synchronous Smartwatch Gestures Using Correlation and Magnetic Sensing,* Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT'18), December 2017, DOI 10.1145/3161162
10. Chen et al., *Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing,* Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI'16.), May 07-12, 2016, DOI 10.1145/2858036.2858125
11. Zhou et al., *AuraSense: Enabling Expressive Around-Smartwatch Interactions with Electric Field Sensing,* Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST'16), October 16-19, 2016, DOI 10.1145/2984511.2984568
12. Zhang et al., *Electrick: Low-Cost Touch Sensing Using Electric Field Tomography,* Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17), May 6-11, 2017, DOI 10.1145/3025453.3025842
13. Seipp et al., *BackPat: One-Handed Off-Screen Patting Gestures,* Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI'14), September 23-26, 2014, DOI 10.1145/2628363.2628396
14. Zhang et al., *BackTap: Robust Four-Point Tapping on the Back of an Off-the-shelf Smartphone,* Proceedings of the adjunct publication of the 26th annual ACM symposium on

User interface software and technology (UIST'13), October 8-11, 2013, DOI 10.1145/2508468.2514735

15. Liang et al., *Deep Learning Based Inference of Private Information Using Embedded Sensors in Smart Devices*, IEEE Network, July 2018, DOI 10.1109/MNET.2018.1700349