December 2021

# Methods for Learning a User's Intent Based on Spatial Gestures

Mingguang Xu

Kai Shi

Ning Zhang

# Methods for Learning a User's Intent Based on Spatial Gestures

**Abstract:**

This publication describes methods for learning a user's intent, when the user interacts with multiple computing devices, based on detected spatial gestures. The methods may utilize a machine-learned model with a reinforcement-learning framework trained to classify spatial gestures, user state information, and/or device state information detected during a device-to-device interaction, generate predictions relating to the intent of the user, and utilize the predictions to facilitate the device-to-device interaction.

**Keywords:**

Machine learning, machine-learned model, reinforcement learning, Q-learning, multi-armed bandits, spatial gesture, user/device state information, user interface, reward, Ultra-Wide Band (UWB) ranging, Bluetooth Low Energy (BLE) ranging, user intent

**Background:**

A spatial gesture is a physical movement of a computing device, by a user, in space, which can be sensed by one or more sensor means of the computing device and/or another computing device. The two computing devices may be configured to interact with one another through what is described herein as a "device-to-device interaction." Spatial gestures are frequently used to trigger device-to-device interactions. For example, by tapping a smartphone against a tablet computer, resulting in a spatial gesture detected by a sensor means (e.g., ranging/distance measurements, proximity sensors) of one or both devices, the smartphone may automatically initiate a media transfer (e.g., transfer an active audio stream) from the smartphone to the tablet computer.

However, problems arise when a detected spatial gesture does not correlate with the actual intent of the user. For example, a user may not wish to transfer media every time their smartphone is tapped against their tablet computer or when a user does want to transfer a media stream from their smartphone positioned a certain distance away from their tablet computer, but only in specific scenarios. The preferences of individual users may prove problematic for designing a single solution for device-to-device interaction based on spatial gestures across multiple devices.

**Description:**

Fig. 1, below, illustrates an example computing device and elements of the computing device that support the disclosed methods for learning a user's intent based on spatial gestures using a machine-learned model with a reinforcement-learning framework (hereinafter just the "ML Model"). The computing device includes one or more processors, a display, and sensor means (e.g., Ultra-Wide Band radar, Bluetooth Low Energy (BLE) triangulation, positioning systems, proximity sensors, accelerometers). The sensor means may be used to calculate a distance between two computing devices (e.g., using ranging, using triangulation).

The computing device also includes a computer-readable medium (CRM). Device data (e.g., applications, a user interface, a reinforcement learning manager, a machine-learned model, and/or an operating system of the mobile device) is stored on the CRM. The machine-learned model predicts user intent based on spatial gestures detected by the sensor means, which are associated with certain device state information and previous user interactions with the device, for example, the user providing feedback (e.g., input on the user interface of the device).

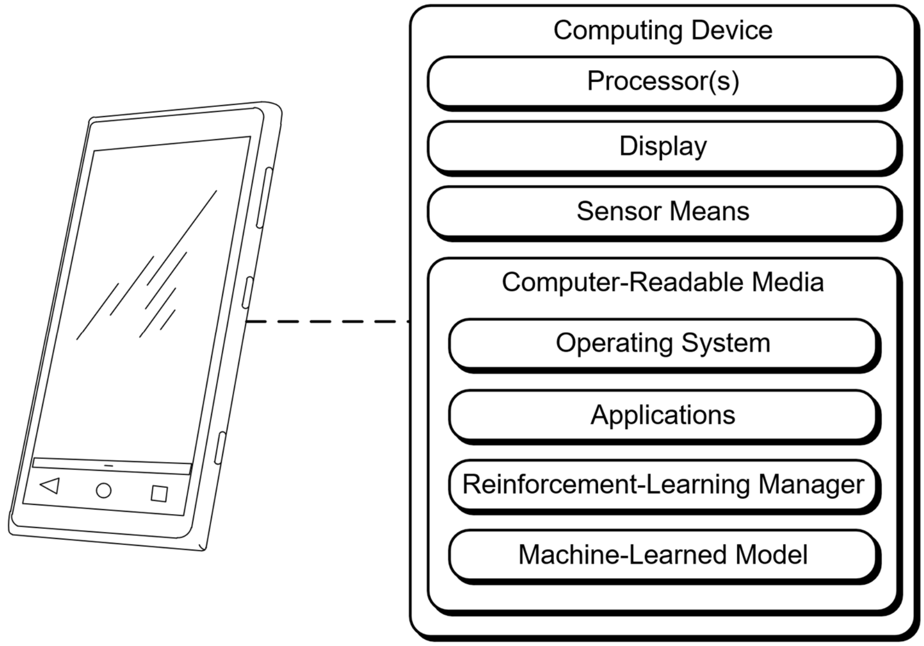


**Fig. 1**

The device data may include instructions that, responsive to execution by the processor, cause the processor to perform operations to implement the described methods. The machine-learned model is trained to classify spatial gestures, user state information, and/or device state information detected during a device-to-device interaction, generate predictions relating to the intent of the user, and utilize the predictions to facilitate the device-to-device interaction.

In this context, reinforcement learning can be leveraged to find a balance between exploration of unknown outcomes (e.g., randomly selecting from a set of variables) and exploitation of known outcomes (e.g., selecting a known quantity from the set of variables). A reinforcement-learning agent interacts with its environment in individual time steps. At each time *t*, the agent receives the current state, the agent then chooses an action from the set of available actions, which is subsequently sent to the environment, and after getting user feedback, the agent can calculate the received reward which is a function of the state, action being taken, and user feedback. The reinforcement-learning agent explores variables to learn a policy for action given the state which maximizes the expected cumulative reward.

In aspects, reinforcement learning may be performed using a Q-Learning algorithm where the objective is to maximize total rewards based on a sequence of actions and user feedback. Assuming the state does not change with the action being taken (e.g., no state transition) the Q-Learning algorithm can update the reward function based on the following formula:

$$Q(s, a) \leftarrow (1 - a) \cdot Q(s, a) + a \cdot r(s, a, \textbf{feedback})$$

where "$Q$" refers to the reward as a function of action "$a$" in state "$s$"; action "$a$" refers to an action (e.g., selection of an element on a user interface, trigger media transfer, no action) by the user on the user interface; state "$s$" refers to elements of the computing device (e.g., time, location, spatial gesture); and reward "$r$" is the immediate reward with action "$a$" and user's feedback on the user interface.

The Q-Learning algorithm provides one approach for modeling a user's intent within the reinforcement-learned framework. The user's intent may vary depending upon the time and location of their device. For example, a user may not wish to initiate media transfer no matter how close the two devices are, while in an office during work hours; however, they may wish to initiate media transfer at a particular time at home outside of work hours. In other aspects, the first computing device may initiate active unlock or call transfer to the second computing device in addition to media transfer. Fig. 2 (below) illustrates an example of how the computing device and/or applications installed on the computing device may use the machine-learned model to predict a user's intent.

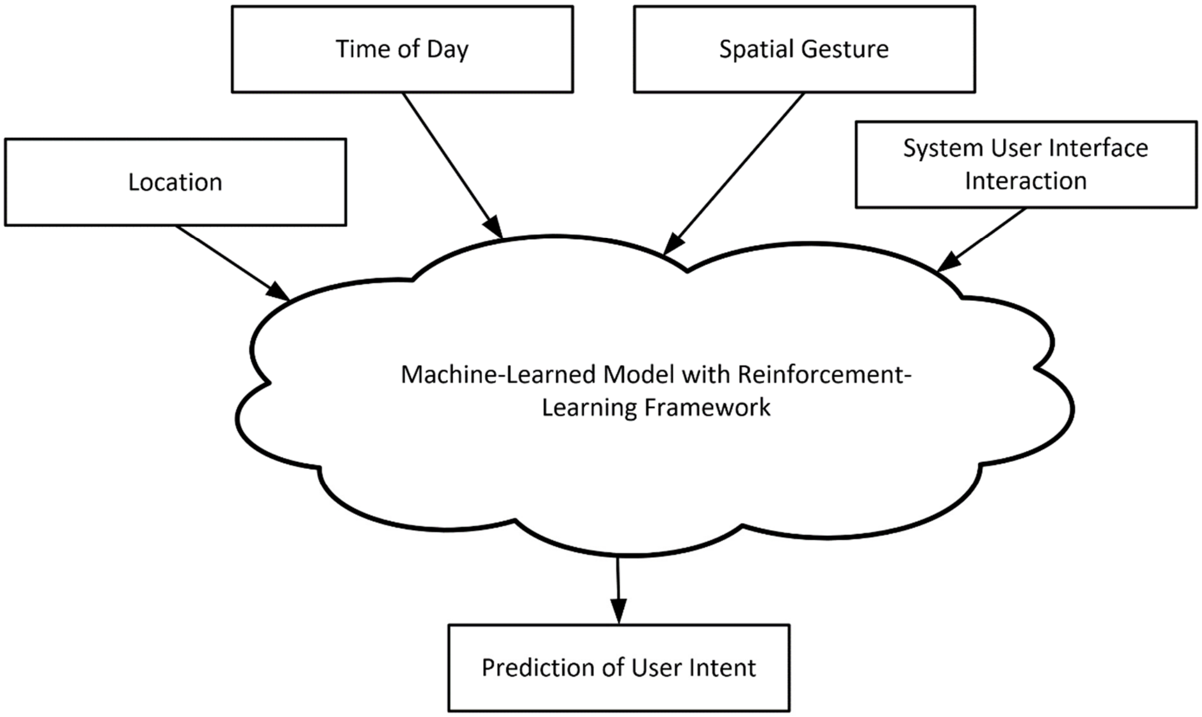


**Fig. 2**

As illustrated in Fig. 2, inputs are provided to the ML Model. Such inputs may include spatial gestures, user state information (e.g., user interface interaction, user feedback), and device state information (e.g., location information, time information, date information).

In an aspect, as a user moves their first computing device closer to a second computing device (e.g., the user makes a spatial gesture by moving their first device within a certain distance threshold of the second device), one or both of the computing devices sense the presence and location of the other computing device (e.g., utilizing sensor means), and a notification (e.g., an element on the user interface) is displayed on the display of one or both of the computing devices. The user may interact with the user interface and provide user feedback (e.g., click on the user interface element) to take an action (e.g., trigger a media transfer). Alternatively, the user may not provide user feedback. The ML Model receives one or more of the spatial gestures, user state information, and device state information and determines a probability of user behavior in each situation. The output of the ML Model is a prediction of future user intent under similar future conditions.

The following use cases describe example scenarios with specific inputs for learning a user's intent, when the user interacts with multiple computing devices, based on detected spatial gestures, utilizing a machine-learned model with a reinforcement-learning framework. Fig. 3 (below) illustrates a first computing device (white smartphone) positioned at a particular distance (e.g., ten feet, four feet) from a second computing device (black tablet computer).

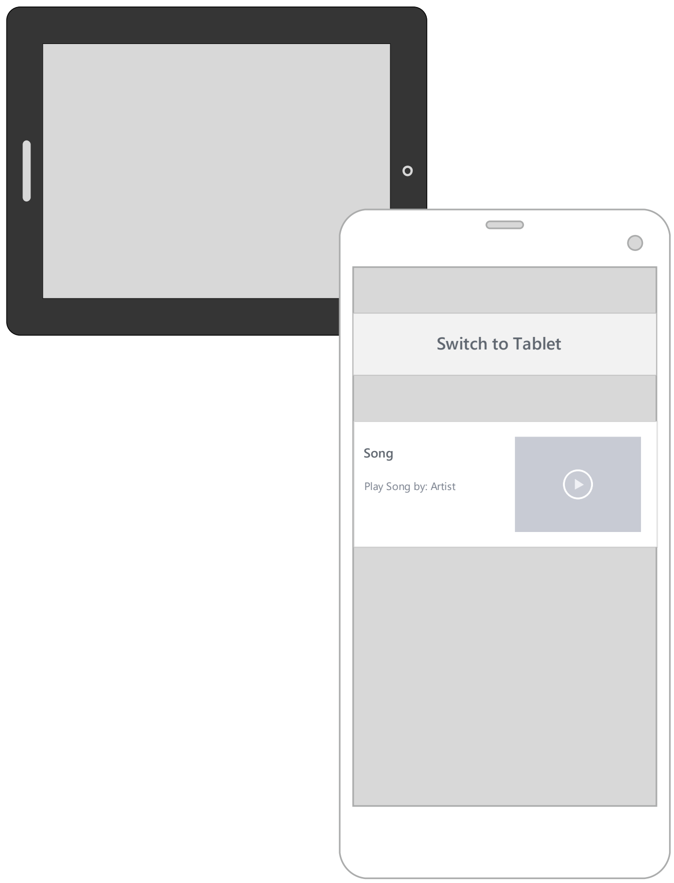


**Fig. 3**

The first computing device senses the second computing device with a sensor means and calculates a distance between the two devices. Upon the first device reaching the particular distance threshold from the second device, a user interface element is displayed on the display of the first device (e.g., a "Switch to Tablet" prompt is displayed for the user) that requests feedback from the user to initiate a media transfer from the first computing device to the second computing device. The user may provide feedback to the device by selecting or dismissing the element. If the user selects the element, the media transfer to the second device will commence (e.g., the audio stream playing on the first device will be transferred to the second device). If the user selects the user interface element, then the ML model learns the user's intent and the user interface element is helpful to the user, and in the future if the same spatial gesture is detected, the same user interface element should be triggered with high chance. On the other hand, if the user never selects the user interface element and always ignores it, then the ML model learns that the user may not have intent to trigger device-to-device interaction, and in the future if the same spatial gesture is detected, the user interface element may not be triggered.

Fig. 4 (below) illustrates the first computing device at a particular distance (e.g., one and one-half feet) from the second computing device. The first computing device senses the second computing device with the sensor means and calculates a distance between the two devices. Upon the first device reaching the particular distance threshold from the second device, a user interface element is triggered on the first device (e.g., a prompt of "Reach Closer to Switch to Tablet" to guide the user to move the first device closer to the second computing device to trigger the device-to-device interaction). If the user follows the user interface element (e.g., the prompt of "Reach Closer to Switch to Tablet") and moves closer to the second computing device, then the ML model learns the user's intent and the user interface element is helpful to the user, and in the future if the same spatial gesture is detected, the same user interface element should be triggered with high chance. On the other hand, if the user does not follow the user interface element and ignores it, then the ML model learns that the user may not have intent to move closer to trigger the device-to-device interaction, and in the future if the same spatial gesture is detected, the user interface element may not be triggered.

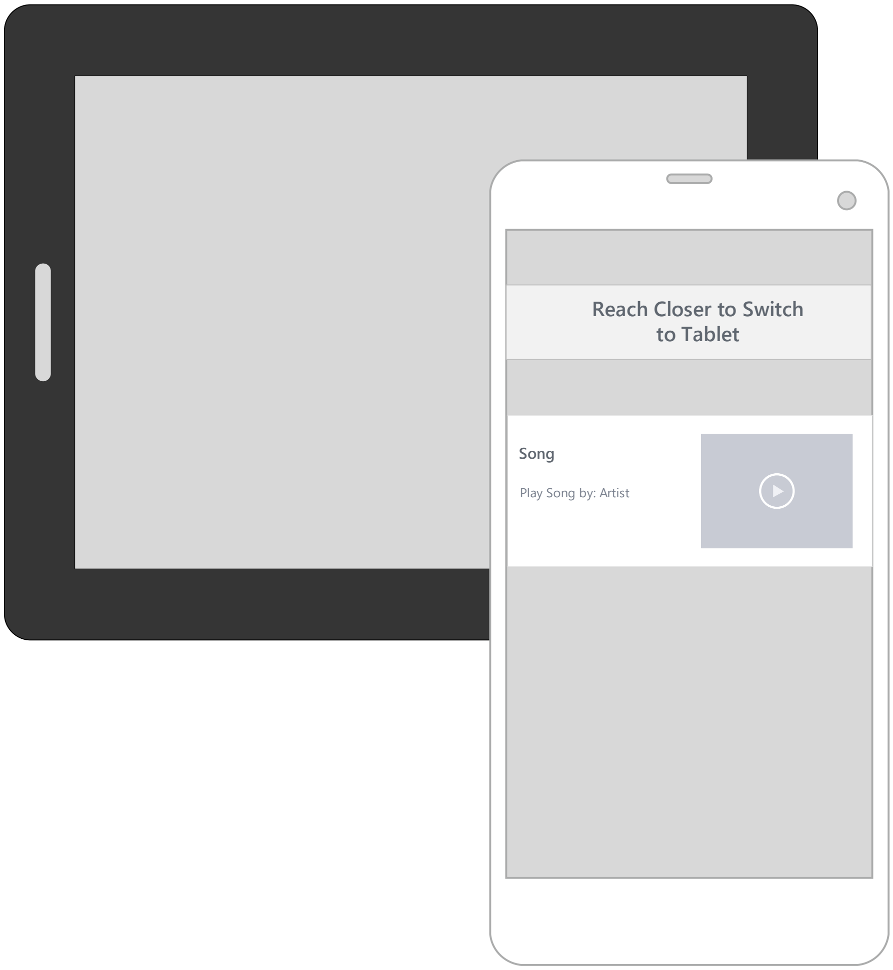


**Fig. 4**

Fig. 5 (below) illustrates the first computing device at a particular distance (e.g., less than three) from the second computing device. The first computing device senses the second computing device with a sensor means and calculates a distance between the two devices. Upon the first device reaching the particular distance threshold from the second device, a user interface element is triggered on the first device (e.g., a prompt to indicate that media transfer from the first computing device to the second computing device commences). After the media transfer, the user of the first device may negate their previous affirmative decision through the selection of an additional user interface element (e.g., “undo”). The user may receive a notification informing the user of a successful media transfer (e.g., a notification “Switched to Tablet”) to the second computing device. If the user does not negate the device-to-device interaction, then the ML model learns the user’s intent and in the future if the same spatial gesture is detected, the same device-to-device interaction should be triggered with high chance. On the other hand, if the user negate the device-to-device interaction by clicking on the “Undo,” then the ML model learns that the user may not have intent to trigger the device-to-device interaction, and in the future if the same spatial gesture is detected, the same device-to-device interaction may not be triggered.

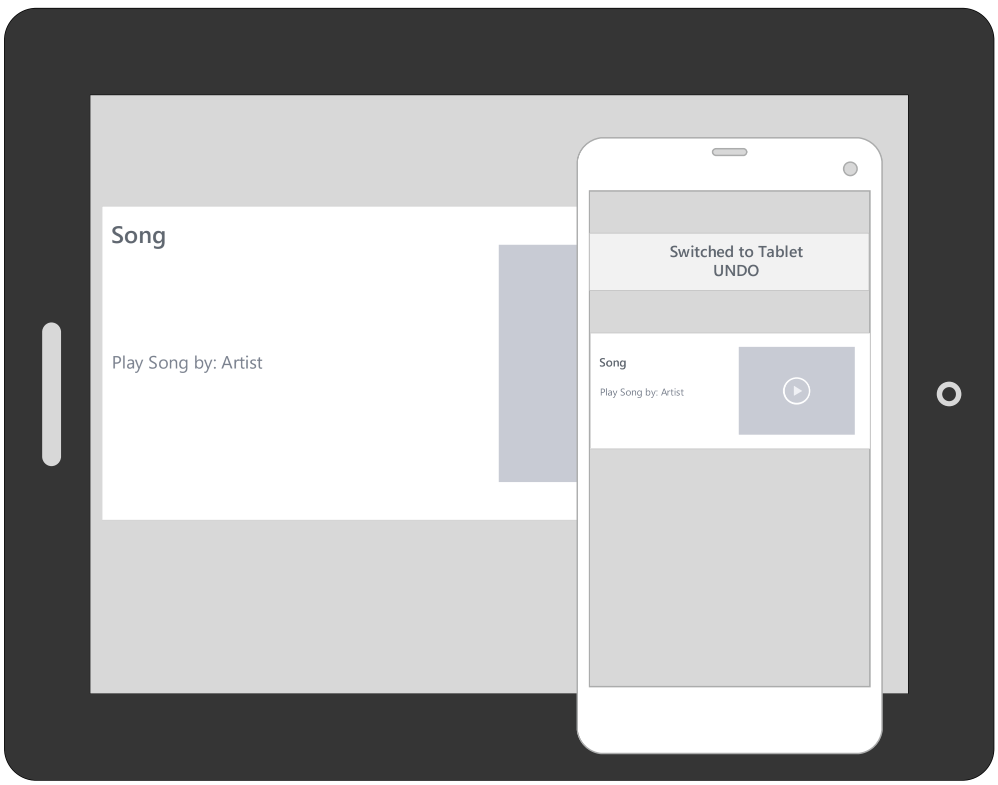


**Fig. 5**

Throughout this disclosure, examples are described where a computing system (e.g., a smartphone, a tablet computer, a computing device) may analyze information (e.g., spatial gestures) associated with a user, for example, the user tapping their smartphone against a tablet computer to initiate media transfer. Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, and/or features described herein may enable collection of information (e.g., information about a user's current location, information about user-specific spatial gestures, user state information, device state information), and if the user is sent content or communications from a server. The computing system can be configured to only use the information after the computing system receives explicit permission from the user of the computing system to use the data. For example, a user may be provided the opportunity to enable or disable location tracking features. Further, individual users may have constant control over what programs can or cannot do with the information. In addition, information collected may be pre-treated in one or more ways before it is transferred, stored, or otherwise used, so that personally identifiable information is removed. For example, a user's geographic location may be generalized where location information is obtained, so that a location of a user cannot be determined. Thus, the user may have control over whether information is collected about the user and the user's device, and how such information, if collected, may be used by the computing device and/or a remote computing system.

**References:**

[1] Patent Publication: CN105892661A. Machine Intelligent Decision-Making Method. Priority Date: March 31, 2016.

[2] Patent Publication: WO2021044586A1. Information Provision Device, Learning Device, Information Provision Method, Learning Method, Information Provision Program, and Learning Program. Priority Date: September 5, 2019.

[3] Patent Publication: US20200103980A1. Systems and Methods for Triggering Actions Based on Touch-Free Gesture Detection. Priority Date: December 13, 2012.

[4] Yang LI, Jin HUANG, Feng TIAN, Hong-An WANG, Guo-Zhong DAI, Gesture interaction in virtual reality, Virtual Reality & Intelligent Hardware, Volume 1, Issue 1, 2019, Pages 84-112, ISSN 2096-5796, https://doi.org/10.3724/SP.J.2096-5796.2018.0006.