November 2022

User Context Aware Notification Delivery

Anonymous

User Context Aware Notification Delivery

ABSTRACT

This disclosure describes techniques related to generating notifications, including smart notifications. The context of a user is captured using an attention tracking model that includes face tracking and/or eye tracking. For example, determining the context can include determining whether a sound is being played to the user. The captured context is provided (continuously, or in a semicontinuous fashion) as an input to a notification service. The notification service us user-customizable and optimizes notifications based on the captured context. For example, optimizing notifications can include generating placement of a notification on a screen of a user device of the user. Optimizing notifications can also include muting a sound.

KEYWORDS

- User context
- User mood
- Notification
- Alert
- Smart notification
- Intelligent notification
- Eye tracking
- Face tracking
- User annoyance
- Machine learning

BACKGROUND

Notifications on a mobile phone or other user device can cause annoyance. One of the reasons for the annoyance is that notifications are often delivered ignoring the context the user is in. For instance, when a user is reading an article on the web, a notification bubble may pop up right where the user is currently focusing their attention. In addition to the “blind placement” of the notification, a superfluous sound may be played even though the user has already seen the notification.

DESCRIPTION

This disclosure describes techniques related to generating notifications, including smart notifications. The context of a user is captured using an attention tracking model that includes face tracking and/or eye tracking. For example, determining the context can include determining whether a sound is being played to the user. The captured context is provided (continuously, or in a semicontinuous fashion) as an input to a notification service. The notification service us user-customizable and optimizes notifications based on the captured context. For example, optimizing notifications can include generating placement of a notification on a screen of a user device of the user. Optimizing notifications can also include muting a sound.

Per the techniques described herein, a machine learning (ML) based solution is provided to reduce superfluous annoyances from notifications. For example, a part of the user context is captured, e.g., specifically where a user's attention is being focused by processing on-device via a front camera stream. A face tracking and/or emotion classifier model is used to determine a mood the user is in. It may be determined whether the user is looking at the screen and/or whether the phone (or another user device) is in a pocket. An eye tracking model can be utilized determine where on the screen the user is looking at. This model may be conditioned on the fact that the user is looking at the screen to save computation. These two models may be provided as inputs to the notification service, e.g., in a semi-continuous fashion, e.g., every second or other suitable time duration to trade-off how fresh the estimates are and battery drainage due to computation.

Provided with these inputs, the notification service (either via an operating system or a specific app running on the user device) can optimize diverse things like placement and whether a sound is being played. For instance, if a notification bubble is bound to be shown at the bottom of the screen where the user is currently reading content in a focused fashion, this will be a disturbing experience. Knowing that the user is focused and where the user's gaze is located, the notification bubble can be shown on the top right corner (or other location) instead, far from the user's gaze location. Further, the notification service may decide not to play a sound. Such preferences can be customized by the user.

The disclosed techniques provide smart notifications based on the user's context and address the technical problem of providing notifications in ways that do not disturb a user while the user is using a device through which the notifications are provided. The techniques can also improve the efficiency in generating notifications.

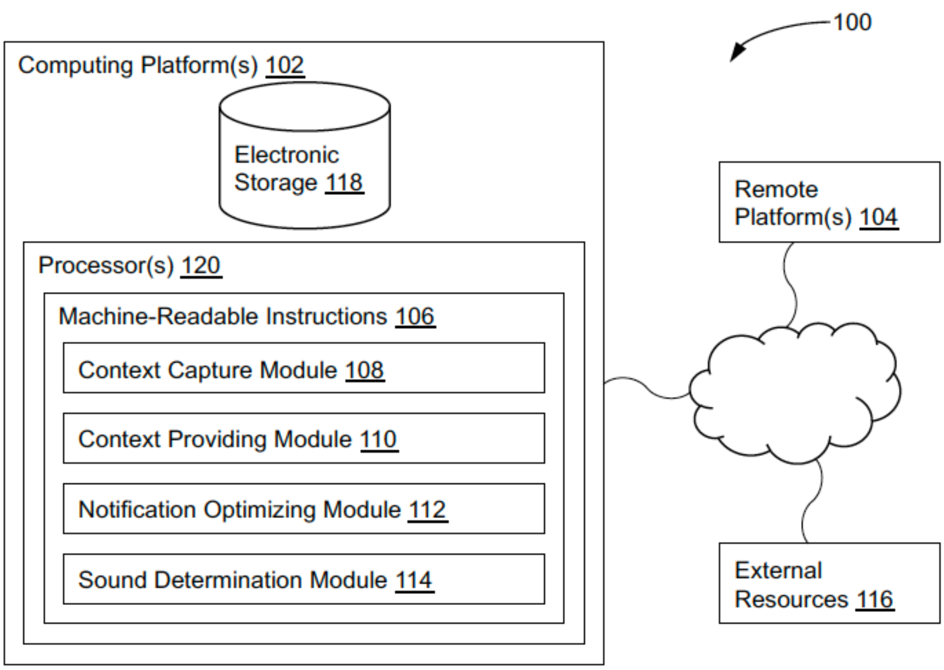

FIG. 1

FIG. 1 illustrates a computer system 100 configured for generating notifications. To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced. System 100 includes one or more computing platforms 102. Computing platform(s) 102 are configured to communicate with one or more remote platforms 104 according to a client/server architecture, a peer-to-peer architecture, and/or other architectures. Users can access system 100 via remote platform(s) 104.

Computing platform(s) 102 are configured by machine-readable instructions 106. Machine-readable instructions 106 may include one or more instruction modules. The instruction modules may include computer program modules. The instruction modules include one or more of context capture module 108, context providing module 110, notification optimizing module 112, sound determination module 114, and/or other instruction modules.

Context capture module 108 is configured to capture a context of a user based on an attention tracking model. The attention tracking model may include face tracking and/or eye tracking. Context providing module 110 is configured to provide the context as an input to a notification service. The notification service may be customizable by the user. Providing the context may be in a semi-continuous fashion.

Notification optimizing module 112 is configured to optimize, by the notification service, notifications based on the context. Optimizing the notifications can include generating placement of a notification on a screen of a user device of the user. Optimizing the notifications can also include muting a sound. Sound determination module 114 is configured to determine whether a sound is being played to the user.

Computing platform(s) 102, remote platform(s) 104, and/or external resources 116 may be operatively linked via one or more electronic communication links. For example, such electronic communication links may be established, at least in part, via a network such as the Internet and/or other networks, or via some other communication media.

A given remote platform 104 may include one or more processors configured to execute computer program modules. The computer program modules may be configured to enable an expert or user associated with the given remote platform 104 to interface with system 100 and/or external resources 116, and/or provide other functionality attributed herein to remote platform(s) 104. By way of non-limiting example, a given remote platform 104 and/or a given computing platform 102 may include one or more of a server, a desktop computer, a laptop computer, a handheld computer, a tablet computing platform, a NetBook, a Smartphone, a gaming console, and/or other computing platforms.

External resources 116 may include sources of information outside of system 100, external entities participating with system 100, and/or other resources. In some implementations, some or all of the functionality attributed herein to external resources 116 may be provided by resources included in system 100.

Computing platform(s) 102 may include electronic storage 118, one or more processors 120, and/or other components. Computing platform(s) 102 may include communication lines, or ports to enable the exchange of information with a network and/or other computing platforms. Illustration of computing platform(s) 102 in FIG. 1 is not intended to be limiting. Computing platform(s) 102 may include hardware, software, and/or firmware components operating together to provide the functionality attributed herein to computing platform(s) 102. For example, computing platform(s) 102 may be implemented by a cloud of computing platforms operating together as computing platform(s) 102.

Electronic storage 118 may comprise non-transitory storage media that electronically stores information. The electronic storage media of electronic storage 118 may include one or both of system storage that is provided integrally (i.e., substantially non-removable) with computing platform(s) 102 and/or removable storage that is removably connectable to computing platform(s) 102 via, for example, a port (e.g., a USB port, a firewire port, etc.) or a drive (e.g., a disk drive, etc.). Electronic storage 118 may include one or more of optically readable storage media (e.g., optical disks, etc.), magnetically readable storage media (e.g., magnetic tape, magnetic hard drive, floppy drive, etc.), electrical charge-based storage media (e.g., EEPROM, RAM, etc.), solid-state storage media (e.g., flash drive, etc.), and/or other electronically readable storage media. Electronic storage 118 may include one or more virtual storage resources (e.g., cloud storage, a virtual private network, and/or other virtual storage resources). Electronic storage 118 may store software algorithms, information determined by processor(s) 120, information received from computing platform(s) 102, information received from remote platform(s) 104, and/or other information that enables computing platform(s) 102 to function as described herein.

Processor(s) 120 may be configured to provide information processing capabilities in computing platform(s) 102. As such, processor(s) 120 may include one or more of a digital processor, an analog processor, a digital circuit designed to process information, an analog circuit designed to process information, a state machine, and/or other mechanisms for electronically processing information. Although processor(s) 120 is shown in FIG. 1 as a single entity, this is for illustrative purposes only. In some implementations, processor(s) 120 may include a plurality of processing units. These processing units may be physically located within the same device, or processor(s) 120 may represent processing functionality of a plurality of devices operating in coordination. Processor(s) 120 may be configured to execute modules 108, 110, 112, and/or 114, and/or other modules. Processor(s) 120 may be configured to execute modules 108, 110, 112, and/or 114, and/or other modules by software; hardware;

firmware; some combination of software, hardware, and/or firmware; and/or other mechanisms for configuring processing capabilities on processor(s) 120. As used herein, the term “module” may refer to any component or set of components that perform the functionality attributed to the module. This may include one or more physical processors during execution of processor readable instructions, the processor readable instructions, circuitry, hardware, storage media, or any other components.

It should be appreciated that although modules 108, 110, 112, and/or 114 are illustrated in FIG. 1 as being implemented within a single processing unit, in implementations in which processor(s) 120 includes multiple processing units, one or more of modules 108, 110, 112, and/or 114 may be implemented remotely from the other modules. The description of the functionality provided by the different modules 108, 110, 112, and/or 114 described below is for illustrative purposes, and is not intended to be limiting, as any of modules 108, 110, 112, and/or 114 may provide more or less functionality than is described. For example, one or more of modules 108, 110, 112, and/or 114 may be eliminated, and some or all of its functionality may be provided by other ones of modules 108, 110, 112, and/or 114. As another example, processor(s) 120 may be configured to execute one or more additional modules that may perform some or all of the functionality attributed below to one of modules 108, 110, 112, and/or 114.

The techniques described herein may be implemented as method(s) that are performed by physical computing device(s); as one or more non-transitory computer-readable storage media storing instructions which, when executed by computing device(s), cause performance of the method(s); or, as physical computing device(s) that are specially configured with a combination of hardware and software that causes performance of the method(s).

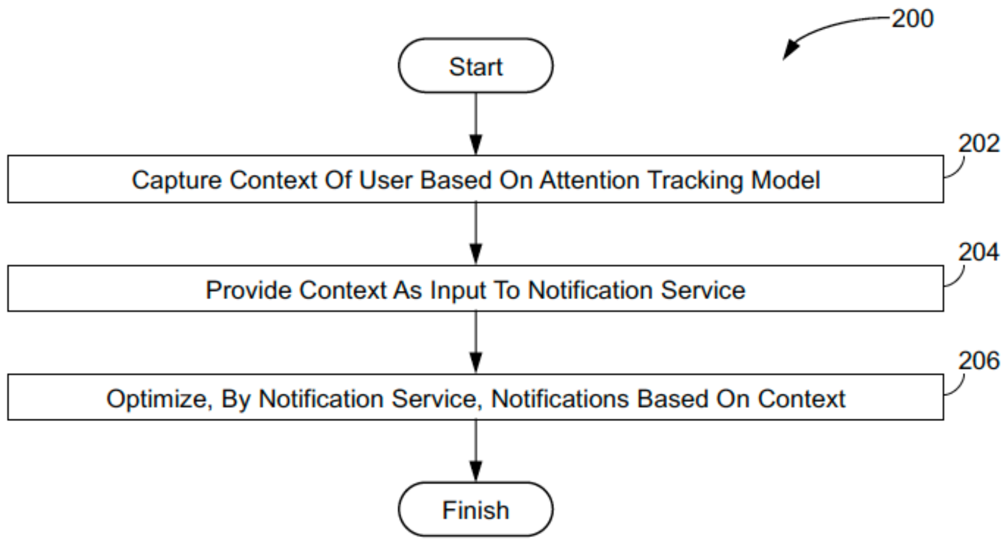

FIG. 2

FIG. 2 illustrates a flow diagram of a method 200 for generating notifications. To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced. For explanatory purposes, the example method 200 is described herein with reference to FIG. 1.

The context of a user is captured (202), e.g., based on an attention tracking model (e.g., by context capture module 108). The attention tracking model can include face tracking and/or eye tracking. example, the context of a user based on an attention tracking model. The context is provided (204) (e.g., by context providing module 110) as an input to a notification service. For example, the context can be provided in a semi-continuous fashion.

The notification service optimizes (206) notifications based on the context, e.g., using notification optimizing module 112. The notification service is customizable by the user. For example, optimizing can include generating placement of a notification on a screen of a user device of the user.

<u>Example of use</u>

For example, determining context can include determining whether a sound is being played to the user. If a sound is being played (e.g., the user is listening to music), optimizing the notifications can include muting the sound, thereby minimizing annoyance to the user.

CONCLUSION

This disclosure describes techniques related to generating notifications, including smart notifications. The context of a user is captured using an attention tracking model that includes face tracking and/or eye tracking. For example, determining the context can include determining whether a sound is being played to the user. The captured context is provided (continuously, or in a semicontinuous fashion) as an input to a notification service. The notification service us user-customizable and optimizes notifications based on the captured context. For example, optimizing notifications can include generating placement of a notification on a screen of a user device of the user. Optimizing notifications can also include muting a sound.