Technical Disclosure Commons

Defensive Publications Series

August 24, 2016

SMART SCREEN ROTATION DETECTION FOR MOBILE DEVICES

Thomas Deselaers

Daniel Keysers

Victor Carbune

SMART SCREEN ROTATION DETECTION FOR MOBILE DEVICES

ABSTRACT

A system and method to identify and implement application-specific screen rotation settings on a mobile are disclosed. The proposed method extends the operating system to use multiple signals, including sensor readings, accelerometer inputs, face detection algorithms, location based signals, or contextual signals as inputs. The rotation setting preferences specific to an application are aggregated based on the multiple signals feeding into a decision algorithm. The decision algorithm is configurable and learns rotation preferences specific to an application or application category. The decision algorithm can implement decisions based on a fixed combination of functions with manually chosen weights for the inputs. The combination of functions can also be learned using machine learning methods such as neural networks, support vector machines, or decision trees. The method further addresses the problem of erroneous rotation of screen content using a smart approach to detect the user's preference.

BACKGROUND

Generally, the screen rotation feature of a mobile operating system does not offer any smart or even basic possibility of storing the rotation or orientation preference by specific application. The problem is further illustrated in the following examples. When using a video application, the user is most likely to visualize a video in landscape mode, while on the other hand to view e-mails, he may prefer to keep the portrait orientation. Even more, in some situations, the orientation preference might change with respect to the position of the user holding the mobile phone. Currently, screen rotation seems to be mainly based on the accelerometer sensor in a system and the preference is set for the entire system layout, and we cannot customize screen rotation required in the situations mentioned earlier. Also, currently, the user can only disable or enable screen rotation for all applications in the settings of the mobile operating system. Thus, there is a need for a smart approach to identify the optimal rotation setting as well as an extension to the existing model to provide configuration settings specific to an application.

DESCRIPTION

The disclosure presents a system and method to identify and implement better rotation settings specific to an application in a mobile phone. It also offers an extension to the existing operating system model for providing the required configuration settings. The proposed method extends the operating system functionality to incorporate other signals and makes screen rotation settings configurable or learnable specific to an application or application category. The rotation settings specific to an application are aggregated based on multiple signals feeding into a decision algorithm. Signals that could be used to improve detecting the user's preference to rotate the device are as follows:

- Currently used: orientation sensor or accelerometer.
- Manually toggling the rotation status should override all other signals.
- A machine learning system could predict from the history of manual toggles of the rotation status if the user wants the phone to be rotated. For example, if a user always uses an application in portrait mode and is now starting the same application, the system would predict that he wants to use portrait mode. In a variant of the learning algorithm, the last used setting for an application could be remembered.

- Face detection: An algorithm using the front-facing camera of the phone could look at the user's face and use the orientation of the face relative to the phone as a signal whether or not to rotate the device.
- Preferred orientation could be specified. Currently, some applications work only in portrait mode and other applications work only in landscape mode. Furthermore, applications that can work in both modes would allow users to specify their preferred orientation. Furthermore, this preferred orientation could be determined automatically by looking at the output of the system across users. For example, a developer of an application might get access to this data or the manufacturer of an application store might provide an interface to inform applications about the current likelihood that a certain application is going to be used in portrait or landscape mode.
- Detecting deliberate or accidental motion of the system. The user may sometimes slowly change his or her position, causing a change in position of the mobile in question. For example, while reading on a sofa the user might slowly move to the side. Currently, at some point a threshold is reached and the screen rotates to a different orientation. Instead of relying only on absolute position information, this information could be combined with temporal information. For example, if a user quickly flips the phone from one orientation to the other, it probably was a deliberate move and the user actually intends the change to happen. However, when the phone is slowly tilted, or has been close to the change threshold for a long time, this can be considered as an accidental motion, and would not trigger the change in orientation of the display.

- By location based or other contextual signals: Users may use their phones differently based on when and where they are using their phones, for example, a user who often reads in the bed before sleeping might intend to use the same orientation independent of their pose. Therefore, signals such as the time ("bedtime") and the location ("bedroom") could be combined with the application for screen orientation.
- Accelerometer inputs: Another temporal signal to use may be the variation of accelerometer inputs over the past (e.g., past several minutes). If accelerometer inputs have changed frequently, the user may be in a situation where he or she is moving (for example, walking or in a car) so another change may more likely be accidental rather than a deliberate attempt to change the screen orientation. If there is access to the state of movement (for example, 'the user is walking'), this would also be used as a signal.

The individual signals as discussed above could be merged to obtain a coherent result. The different input signals are used in an algorithm to come up with a decision on rotating the screen. This can be done in various ways as indicated below.

- A fixed combination of functions with manually chosen weights. For example, a set of rules of the form 'if screen rotation is disabled for this particular application, then do not rotate independent of the other signals' or a combination function that takes a weighted combination like 'use 50% of the signal from the gravity sensor, 30% from a per application learned preference and 20% from a per application type configured value'.
- The combination function could be learned from examples using machine learning (for example, neural networks, support vector machines, decision trees or the like). In this case, a set of examples with input signals and desired output (should the screen be rotated

or not) could be collected, and a machine learning system would be applied to predict the screen rotation status. The examples could be combined using a set of anonymized samples from a large set of users/apps and a set of samples from the user(s) of the particular mobile device.

- An important part of the merging algorithm could also be to guarantee temporal smoothness. For example, it is undesirable for the screen to be constantly switching between portrait and landscape mode, and a certain delay could be inserted between such screen rotations. This temporal delay may depend on the likelihood of how certain a switch is desired, for example, if the local prediction says: “switch with probability 0.6” this could be ignored if a switch just happened before, however if the local prediction says “switch with probability 0.9999” then the switch should probably happen, even if the last switch is only a second ago.

Alternative implementations of the method disclosed might use a subset of the signals or a different machine learning method or combination scheme compared to those discussed here. The method thus addresses the problem of erroneous rotation of screen content in mobile phones using a smart approach to detect the user’s preference and provides a more customized and improved user experience.