

Technical Disclosure Commons

Defensive Publications Series

August 22, 2017

Adaptive Ui Optimization Based On Automatic Detection Of Usage And Orientation Of A Mobile Phone

Sandro Feuz

Thomas Deselaers

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Feuz, Sandro and Deselaers, Thomas, "Adaptive Ui Optimization Based On Automatic Detection Of Usage And Orientation Of A Mobile Phone", Technical Disclosure Commons, (August 22, 2017)
http://www.tdcommons.org/dpubs_series/647



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

ADAPTIVE UI OPTIMIZATION BASED ON AUTOMATIC DETECTION OF USAGE AND ORIENTATION OF A MOBILE PHONE

ABSTRACT

A system and method are disclosed that enable adaptive user interface (UI) optimization by automatically detecting usage and orientation of a mobile phone. The method includes detecting how the phone is oriented (portrait/landscape orientation) and by which hand (right or left) the phone is held and by which hand it is operated by the user. The input data such as raw touch input data, raw gyro sensor data, context of the phone (active application) or context of the user (time of day) are collected from the phone sensors and fed to a machine learning (ML) model. The ML model infers an easily accessible screen area. The system then optimizes the user interface to have the most relevant touch points or controls within the easily accessible area. The optimized adaptive UI is then presented to the user.

BACKGROUND

At present, mobile phones have screen sizes which the user may find difficult to hold in one hand while still being able to reach the whole screen. Instead, depending on how (portrait/landscape orientation and right or left hand) the phone is being held and which hand the user has free, the easily accessible part of the screen differs. Similarly, the easily accessible areas change position when using the phone in landscape orientation. For instance, a user may be viewing his emails on his mobile phone device while walking to the bus and holding one's morning coffee in the left hand and the phone in the right hand. The relevant buttons for reading emails (archive, mark-as-unread, trash, reply) may be located such that they may not be reachable when the user tries to operate with just a single hand on a larger (e.g. 5 inch+) screen.

DESCRIPTION

A system and method are disclosed that enable adaptive user interface (UI) optimization by automatically detecting usage and orientation of a mobile phone using a machine learning (ML) model. The system includes a user device which may be a mobile phone or a tablet with optimized UI. The method to enable optimized adaptive UI on mobile phone is shown in FIG. 1. The method includes detecting how (portrait/landscape orientation and right or left hand) the mobile phone is held by the user. The method in step **101** includes collecting input data such as raw touch input data, raw gyro sensor data, context of the phone (active application) or context of the user (time of day). Collected inputs at step **101** are then sent to the ML model at step **103**. The inputs may determine which hand is used to operate the phone, which hand is used to hold the phone, and the orientation (portrait or landscape) of the phone, which are used to infer an easily accessible screen area at step **105**. The UI of the app is then optimized to locate the controls within the accessible area at step **107**. The optimized adaptive UI is presented to the user at step **109**.

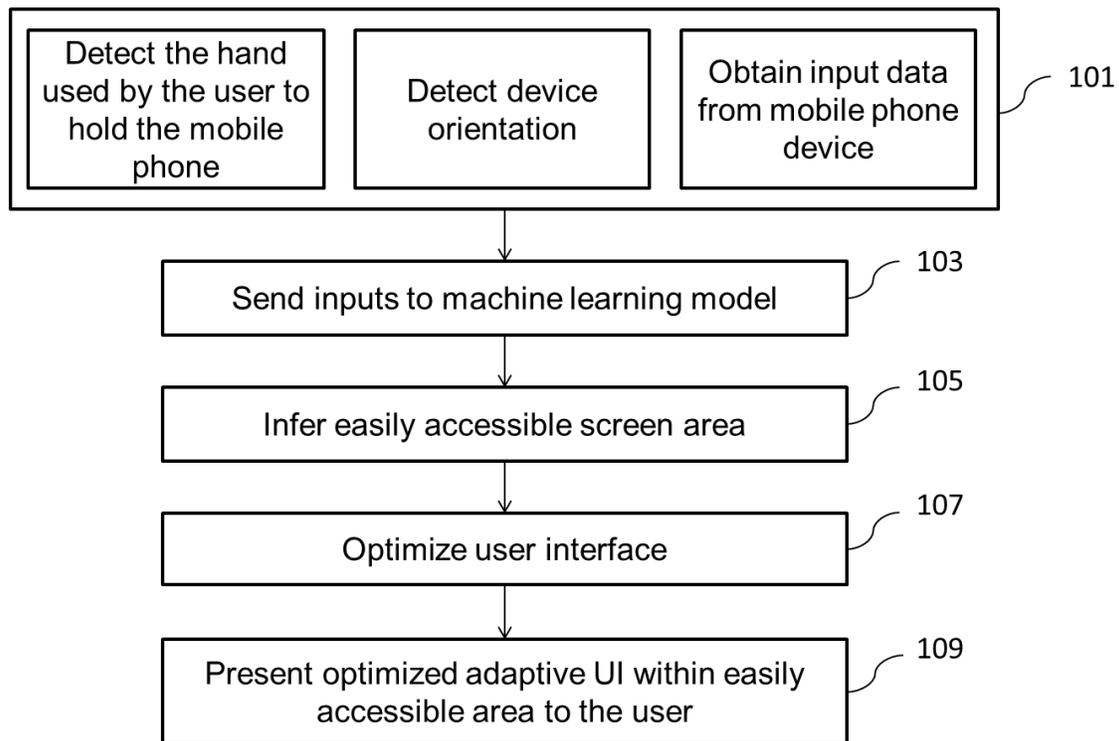


FIG. 1: Method to enable optimized adaptive UI on a mobile phone

The first step **101** involves detecting how (portrait/landscape orientation and right or left hand) the hand (left or right) the mobile phone is held by the user. The method may include determining a pre-defined set of combinations of orientation of the mobile and hand holding the phone that would determine the accessible area. For instance, with the phone in portrait mode, the user may hold on the bottom portion with the right hand and may use touch screen with the right thumb. When the phone is in landscape mode, the user may hold on the right side with the right hand and may use the right thumb to access touch screen. With the phone in landscape mode, the user may hold the middle of the phone with the right hand and may use the right thumb to access the touch screen. With the phone in portrait mode, the user may hold the bottom with the right hand and access the touch screen with the left index finger and the like. Through all the symmetric options about ten to fifteen classes are obtained.

The machine learning model may in step **105** use the following inputs among others to infer the useful screen area: raw touch input data, raw gyro sensor data, context of the phone (active application) or context of the user (time of day). The ML model may be a deep feed forward model extracting features from the raw input through convolutional layers, or a recurrent neural network (RNN) model (such as an LSTM or GRU model) which may take the sensory data as a natural time series input.

Further, training data for the ML model may be collected by allowing the user to perform click-actions while holding and using the phone in a given orientation. The user is directed to click on a square which is placed on the screen. The phone sensors and the necessary data used to compute the aforementioned inputs may be recorded by the ML model. Each training sample may then consist of sets $\langle \text{orientation, hand, data} \rangle$ where orientation encodes the device

orientation (portrait/landscape), hand encodes which hand the user is holding and operating the phone with (left/right/both) and <Integer (class index how (portrait/landscape orientation and right or left hand) the hand (left or right) and usage from the list above), sensor and input data>. The obtained data is used to train the ML model. Additionally, to just predict the class based on orientation of the mobile phone (how the phone is held) and would be used, a certainty score for each class may be generated as an output.

The method may involve dynamically applying the ML model to evaluate changes in the way the phone is being held and the accessible area may be evaluated and predicted at regular intervals. With the model outputting a certainty score, the method may employ a threshold and only act on a specific change if the model is “certain enough” that the useful screen area would change.

Whenever the ML model detects (with a higher level of certainty) that the user has switched to position the phone in a new way, the system and method may act in one of the following ways depending on the context. If the currently active app implements the system API to optimize the UI for how the user is now holding and using the phone, the system provides that information to the app and allows it to redraw its layout. The control and responsibility is then given to the app. Where apps may not support the disclosed API or for system contexts to take an advantage of the new functionality, the system may automatically adjust the (important) UI elements of the active app to make them easier to reach in the current configuration.

The automatic ways to adjust the UI (2) need some knowledge about the easily reachable areas of the screen for the current mode of operation of the mobile phone (the class from above). This may either be obtained by just pre-defining an area for each class, the example of which is shown in FIG. 2.

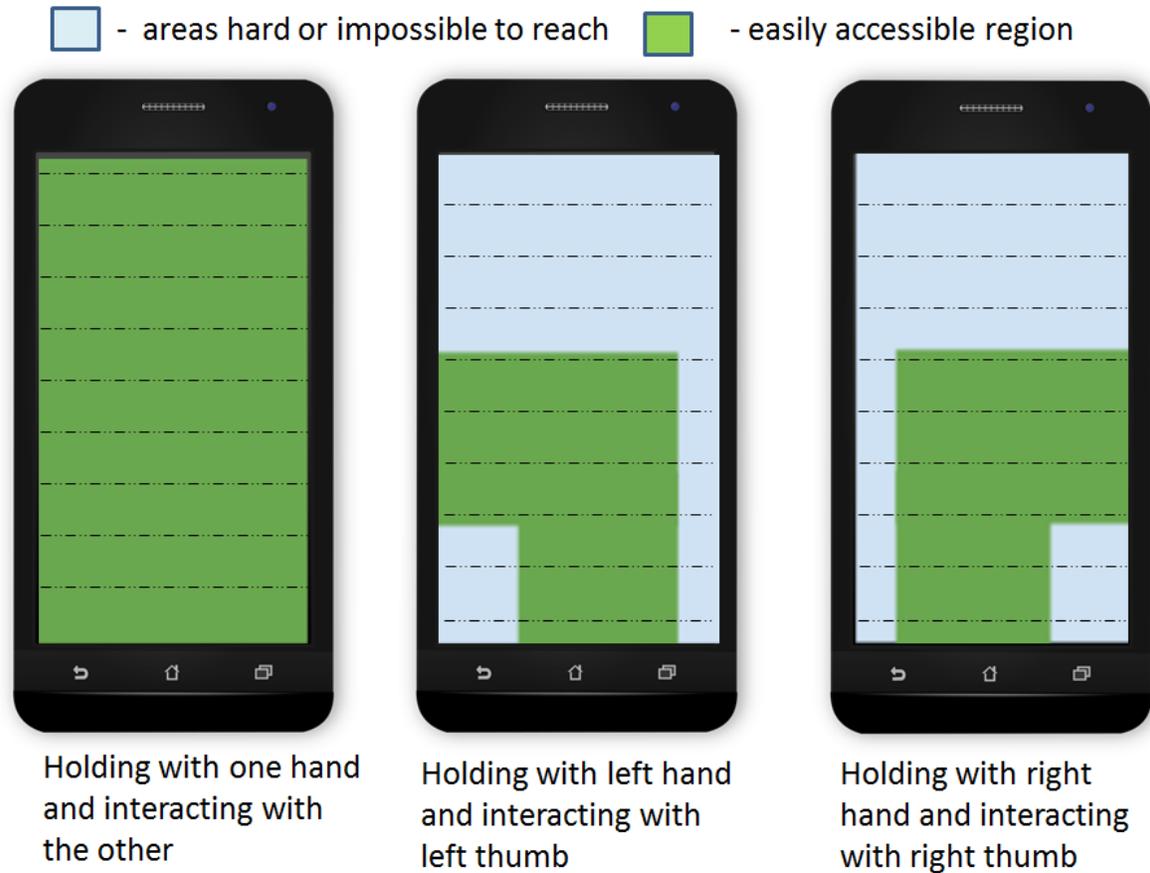


FIG. 2: Optimized adaptive UI on mobile phone for one-handed operation

Also, the method may allow the user to personalize the access area in the different classes by having the user touch all the areas of the screen which the user may easily reach. This would allow for smaller and larger screens and smaller and larger hands that may mean large differences in the easily accessible area in a given mode.

Given the easily accessible area, the UI may be optimized in step **107** in one of the following ways: scaling the whole app so that most of the UI (important) elements may be present in the easily reachable area, scale single (important) UI elements (e.g. buttons) to make them overlap the easily reachable area, move (important) UI elements to the easily accessible area, or overlay a very simplified (potentially partially transparent) UI of the important UI elements in the easily reachable area. To obtain a measure of the importance of single UI

elements, simple statistics may be maintained. This is estimated on how often which UI element is used or train a small ML model which predicts the click-probability of each UI element given the current context of the active app.

In some aspects, the machine learning may be replaceable by either manual user input (needing more manual user interactions which may be hard depending on the mode), or heuristics-given.

Although the disclosed system and method may work in a particular way, the method provides for an app to customize the way the method is implemented. Allowing the system to provide a simplified layout for different modes may reduce the need for app developers to individually customize their apps.