A Novel System for Processing User Interfaces

Frank Edughom Ekpar

***Abstract*—In this paper we introduce novel automatic and manual processing systems for a versatile graphical user interface comprising one or more N-dimensional background elements each of which is divided into one or more arbitrarily-shaped N-dimensional partitions, wherein each partition may contain one or more user interface elements and is associated with one or more sets of rules that define rendering, positioning, element placement and other relevant attributes and behaviors, wherein said rules can be specified in such a way as to enable said N-dimensional background to assume any desired arbitrary shape and to facilitate expansion to any desired arbitrary size without distortion or loss in quality. Our systems permit the efficient generation of intuitive graphical user interfaces in a wide variety of application domains.**

***Index Terms*—Automatic Partitioning, Manual Partitioning, User Interface Processing, Arbitrarily-Sized User Interfaces, Arbitrarily-Shaped User Interfaces, Lossless Resizing, Intuitive User Interface.**

I. Introduction

Contemporary graphical user interfaces are limited in that when they allow arbitrary shapes, they are generally not expandable and when they are expandable, they generally do not permit the use of arbitrary shapes. Furthermore, these graphical user interfaces are generally limited to flat 2-dimensional or at best simulated 3-dimensional structures. Popular graphical user interface systems from software developers such as Microsoft and Apple suffer from these limitations.

Liu et al. [1] teach a graphical user interface that could be stretched or resized. The characteristics or nature of designated areas (such as "border" or "resize" regions) could be used to provide hints (such as a change in the shape of the cursor) to the user that resizing or stretching is possible or occurring at a particular position. However, Liu et al. fail to teach the use of adaptive or selective rendering of the areas of the graphical user interface to achieve the said resizing or stretching. In fact, Liu et al. fail to teach any specific way to achieve the resizing or stretching at all. Furthermore, Liu et al. teach "creating one or more first regions . . . " and "creating one or more second regions" for the user interface. Thus, Liu et al. teach the use of at least two regions--at least one first region and at least one second region.

N. M. Kishore et al. [2] introduce a graphical user interface (GUI) based on GUI objects but fail to teach any specific way of resizing the user interface.

Hamlet et al. [3] describe the use of "optimized vector image data" to enable the display of a graphical user interface in any shape and at any size with minimal or no loss of original image quality, Hamlet et al. fail to provide the option of utilizing the nature (related to the appearance or texture) of the original image. In fact, Hamlet et al. teach away from the use of the appearance of the original image. However, according to the principles of the present invention, the adaptive or selective application of appropriate rules to selected regions of the original image based on the nature (related to the texture or appearance of same) can achieve "infinite resolution" and permit the user interface to assume any desired shape and size without easily noticeable distortion or loss of quality.

Additionally, it should be noted that according to the teachings of Hamlet et al., the "optimized vector image data" is generally created separately and could also be stored separately from the graphical user interface to which it is applied and that changes in certain attributes (such as size) of the interface may necessitate re-computation of vector data. In contrast, the present invention provides the option of directly using the original image and adaptively applying appropriate rules to selected regions of the image with suitable characteristics to facilitate user interfaces that can assume any size and shape. Thus, the options provided by the present invention obviate the need to generate, access or otherwise compute or re-compute vector data, leading to savings in resources and allowing for faster and more responsive and richer user interfaces than permitted by the prior art.

Callaghan et al. [4] elucidate the workings of a graphical user interface that employs scalable vector graphics (SVG) for rendering--including scaling, resizing or stretching. Similarly, Kaasila et al. [5] describe a graphical user interface that utilizes a plurality of scale factors for scaling, resizing or stretching. It should be noted that the scale factors utilized by Kaasila et al. can be selected from a list of available scale factors or generated in response to user interaction with the user interface. In the same vein, Rimmer et al. [6] introduce systems and methods for automatically determining a content item size which may be based on a size of a viewport and a width of a parent element.

None of the prior art teaches a user interface wherein resizing is based on the nature of selected regions.

Similarly, none of the prior art teaches a user interface wherein one or more design characteristics of one or more partitions are guided by one or more intended presentation characteristics of the affected partition. This work improves on an earlier version by F. E. Ekpar [7] by specifying algorithms and systems for automatic and manual processing.

It is an objective of this work to overcome the limitations of the prior art set forth above by providing a versatile graphical user interface comprising one or more N-dimensional background elements each of which is divided

Published on January 15, 2020.
F. E Ekpar is with Admiralty University of Nigeria, Ibusa/Ogwashi-uku Expressway, Delta State, Nigeria. (e-mail: frankekparng@gmail.com)

into one or more arbitrarily-shaped N-dimensional partitions, wherein each partition is associated with one or more sets of rendering, positioning, element placement and other relevant rules and may contain one or more user interface elements--thus enabling said N-dimensional background to assume any desired arbitrary shape and to facilitate expansion to any desired arbitrary size without distortion or loss in quality. This leads to much more versatile, more dynamic and richer user interfaces than are possible with the prior art.

This paper also discloses a system for the automatic or manual processing of the user interface to facilitate dynamic resizing of the interface.

The remainder of this paper is organized as follows. Section II introduces the system design and highlights implementation routes. In Section III, the algorithms for automatic and manual partitioning of the background of the user interface are described. Section IV demonstrates how rules are assigned to identified partitions and how the partitioning scheme is then applied to the background to yield the versatile user interfaces described herein while Section V contains concluding remarks.

II. System Design and Implementation

Generally, a computer system such as a personal computer system, workstation, server, tablet computer system, handheld or mobile computer system and any other suitable system could be used to embody the present invention. Other suitable devices and systems providing means for or allowing the steps of the present invention to be carried out could be used. When a computer system is used, user interaction with the user interface could be via a mouse or any other suitable means. Data for the interface could be stored in computer memory and software running on the computer system could be used to allow editing and presentation of the user interface. The user interface could be presented or rendered on a computer monitor or screen. Suitable computer network systems could be used to implement and/or present aspects of the user interface.

Referring now to Fig. 1, an illustration of a preferred embodiment of the present invention, the arbitrarily-sized and arbitrarily-shaped background is indicated generally as B. In Fig. 1, P1, P2, P3,. . . , PK are partitions. K can be any number. For simplicity, the background and partitions in Fig. 1 are 2-dimensional. Furthermore, the partitions can be contiguous or non-contiguous. In practice, however, the background and partitions are N-dimensional (where N can be 1, 2, 3, 4--for 3 spatial dimensions and 1 temporal dimension for instance, 5, or any number of dimensions) and the partitions need not be contiguous. Furthermore, the partitions need not be literal--in which case a background comprising a two-dimensional image would need to be broken up into a plurality of images to support a plurality of partitions--but could be logical or conceptual only--in which case said background image could remain monolithic. Each partition may contain any number of user interface elements. According to the principles of the present invention, each partition has an arbitrary shape and an arbitrary size and is associated with a set of rules that define rendering, positioning, element placement and other relevant behaviors and attributes. (Generally, attributes and behaviors or characteristics or aspects of the user interface are chosen on the basis of usefulness or relevance in a given embodiment.)

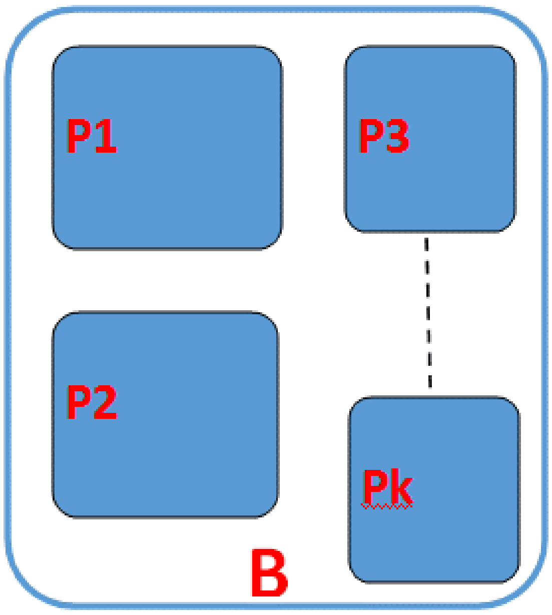

Fig. 1. Partitioning of the background

These rules can be specified in such a way that the N-dimensional background-based graphical user interface can assume any arbitrary desired shape and can be expanded to any arbitrary desired size without distortion or loss in quality. For instance, if the background comprises a single, arbitrarily-shaped digital image and the user interface built from said background is to be rendered on a computer screen, then said background can be divided into a number of partitions based on the nature of the background and the rendering of each partition can in turn be carried out on the basis of the nature of the partition. A partition defined on a uniformly textured region of the background can be stretched without noticeable distortion or loss in quality. In contrast, a partition defined on a non-uniform region of the background may be rendered in its original size and shape to prevent distortion and loss in quality. By creating a number of partitions based on the nature of the background and selectively assigning appropriate sets of rules for rendering, positioning, component placement and other behaviors and attributes of each partition, the entire background can be made to assume an arbitrary shape and an arbitrary size without distortion or loss in quality. Consequently, user interfaces based on the principles of the present invention are more versatile, more dynamic and allow a much richer user experience than is possible with the prior art.

III. Automatic and Manual Partitioning

Fig. 2 shows a flowchart for automatic processing of the user interface. The user interface could comprise a background image that is to be partitioned. Partitioning could be based on the nature of the image. For example, the texture of the image could be used.

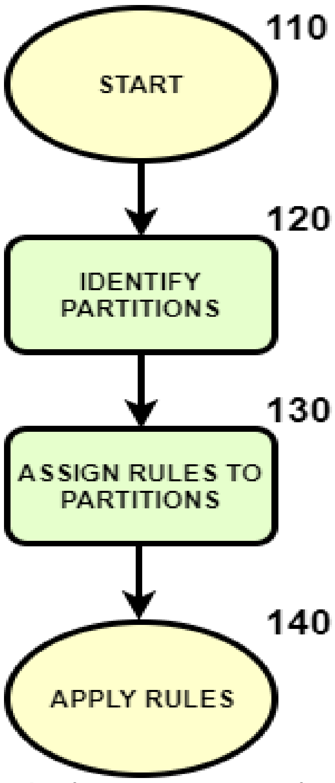

Fig. 2. Flowchart for automatic processing of user interface.

In the step indicated generally as 110 (START) in Fig. 2, the image could be prepared for processing. Such preparation could involve storing the image in memory and providing means of accessing the data representing the image. It could also comprise--in the case of a network-based system--the streaming or transmission of the data representing the image for further manipulation. If required, preparation could also involve pre-processing steps such as filtering and de-noising of the image or the application of any combination of any required pre-processing steps as is well-known in the art.

In the step labeled 120 (IDENTIFY PARTITIONS), an automatic process for the identification of distinct partitions or regions within the image could be carried out. The identification could be based on the texture of the image. Image segmentation techniques (including those popular in the literature) could be used in this step. Any other suitable process for automatically identifying partitions could also be used. The process could be completely automated--in which case the results of an automatic image processing step such as image segmentation are used as the basis for partitioning the image. Furthermore, a semi-automatic process could be used--in which case the automatic partition identification process could be augmented (via user or designer inspection) to manually correct any misidentifications or to more closely conform to user taste. More generally, the step of identifying partitions in the user interface comprises assigning a label to each element in the image such that elements with the same label share common characteristics. In the case of a digital image, each such element would be a pixel. For simplicity, the texture of the image could be chosen as the characteristic on which the labeling of elements is based. It should be noted that any other suitable characteristic (including, but not limited to, shape) could be chosen as the basis of the partitioning. Typical labels could be SIMPLE (for elements with a simple or uniform texture), COMPLEX (for elements with a relatively more complex texture), HORIZONTAL (for elements with a texture that appears horizontal) and VERTICAL (for elements with a texture that might appear vertical). Other suitable labels could be used. For simplicity, the boundaries of identified partitions could be expanded to abut neighboring partitions when necessary in order to ensure that all identified partitions taken together cover the entire background without gaps.

According to the principles of the present invention, the partitioning process is not limited to the automatic and/or semi-automatic processes described in the foregoing. Partitioning could be carried out manually on the basis of a visual inspection of the user interface. Manual identification of partitions could be accomplished on a computer system via suitable instructions (possibly embodied in application software) that permit the designer or user to identify and/or label partitions. This could be accomplished by clicking and dragging a computer mouse over the background image to demarcate or identify and/or label partitions. The labels (for example SIMPLE, COMPLEX, HORIZONTAL, VERTICAL, etc) mentioned for automatic and/or semi-automatic partition identification could also be applied to manual partition identification.

IV. Assigning Rules and Applying the Partitioning Scheme

In step 130 (ASSIGN RULES TO PARTITIONS), each partition identified in step 120 could be associated with a set of rules defining the characteristics of the partition. For example, based on the texture of the identified partition, a specific partition could be designated for vertical tiling during rendering or presentation. By way of example, consider the situation in which the rendering or presentation of the user interface is the characteristic that is to be defined. Any partition labeled SIMPLE could be assigned a rendering or presentation rule that effectively causes the partition to be stretched to fit its destination. In the case of a two-dimensional digital image, this could be accomplished via simple two-dimensional (horizontal and vertical) pixel replication as is well known in the field. In contrast, a partition labeled COMPLEX could be assigned a rendering or presentation rule that effectively causes the partition to be rendered at its actual size--in which case any destination region allocated to the partition could be constrained to the same size and shape as the original partition. A partition labeled HORIZONTAL could be assigned a rendering or presentation rule that effectively causes the partition to be tiled horizontally while a partition labeled VERTICAL could be assigned a rendering or presentation rule that effectively causes be partition to be tiled vertically during rendering or presentation on a destination surface or device.

One of ordinary skill in the art would appreciate that it is possible to synthesize a mapping or table associating rules or sets of rules for rendering (or any other chosen characteristic) with a label or sets of labels identifying partitions within the interface.

Finally, assigned rules can be applied in step 140 (APPLY RULES) to the associated partitions in the storage, presentation and/or more generally further manipulation of the user interface.

Additionally, identification of partitions and/or assignment of rules to identified partitions could also be based on a selected user or designer profile. Such a profile could be built up automatically on the basis of prior user interaction with the user interface, user preferences or other relevant data gathered about the user or designer. For example, a specific user or designer could prefer that SIMPLE partitions be tiled vertically while another could prefer that such partitions be simply stretched via pixel replication or an equivalent process. Via appropriate program code or computer software, the user or designer could be permitted to edit such a profile or build a new profile from scratch. Thus, user or designer profiles could be built explicitly on the basis of user or designer input. Another option is to use a semi-automatic approach in which a user or designer profile could first be built automatically (possibly on the basis of the known behavior and/or preferences of a wide spectrum of users or designers or via some other suitable means) and the automatically synthesized profile subjected to optional editing by users or designers.

The background, partitions, associated user interface elements, user or designer profiles and any required configuration information could be managed as elements in a universal file format. Such a universal file format would specify a header identifying the file type and containing information as to the number, types, locations and sizes of the elements it contains. Each element in the file is in turn described by a header specifying the type of the element, its size and any relevant data or attributes and the types, locations and sizes of any additional elements it contains. By making use of self-describing elements in the manner explained in the foregoing, the universal file format would be able to store an arbitrary element having an arbitrary number and types of other such elements embedded in it.

V. Conclusion

We have introduced novel automatic and manual processing systems for a versatile graphical user interface that enable an N-dimensional background to assume any desired arbitrary shape and to be expanded to any desired arbitrary size without distortion or loss in quality, giving rise to the efficient generation of intuitive graphical user interfaces that can be utilized in a wide variety of application domains.

Our systems lead to much more versatile, more dynamic and richer user interfaces than are possible with the prior art.

It should be understood that numerous alternative embodiments and equivalents of the invention described herein may be employed in practicing the invention. Thus, it is intended that the appended claims define the scope of the invention and that methods and structures within the scope of these claims and their equivalents be covered thereby.

References

[1] S. Liu and L. M. Weitzman, “Modeless interaction with GUI widget applications,” *United States Patent Application Number* 10/769002, 2005.

[2] N. M. Kishore, J. Nikhil, M. Debi and A. J. Samuel, “Cursor tracking in a multi-level GUI,” *United States Patent Number 7,165,225*, 2007.

[3] T. Hamlet and R. Umbehant, “Infinite resolution scheme for graphical user interface object,” *United States Patent Number 6,606,103*, 2003.

[4] M. D. Callaghan and A. B. Batke, “Systems and methods that utilize scalable vector graphics to provide web-based visualization of a device,” *United States Patent Application Number 10/731940*, 2007.

[5] S. J. Kaasila and J. S. Collins, “Methods, systems, and programming for computer display of images, text and/or digital content,” *United States Patent Number 7,222,306*, 2007.

[6] G. J. Rimmer and L. J. Hemens, “Automatically determining a size for a content item for a web page,” *United States Patent Number 10,445,406*, 2019.

[7] F. E. Ekpar, "Versatile user interface," *United States Patent Application Number 11/097879*, 2005.