A PROTOTYPE CARTOGRAPHIC USER INTERFACE FOR WEARABLE COMPUTING

Clarke, K.C.[1], Du, Q.[2], Nuernberger, A.[1] and Pingel, T.[1]

[1]Department of Geography-NCGIA.University of California,
Santa Barbara, CA 93106-4060, USA.
[2]School of Resource and Environmental Science, Wuhan University,
430072 Wuhan. PR China.

ABSTRACT

There is currently no human-computer interface specifically designed for the alternative interaction modes used on wearable systems for input and output. For a geographically oriented wearable computer system a graphical user interface (GUI) is indispensable, yet contemporary window systems using the desktop metaphor are unsuitable for wearable computing. In this paper we document and discuss the design and implementation of a prototype cartographic user interface for wearable computing focused on cartographic tasks, particularly navigation in the field. This system can accommodate multiple sources of spatial information including maps, imagery and thematic data collected in the field, and exploits the two most basic advantages of wearable devices, real-time visualization and multi-modal interaction.

1. INTRODUCTION

Project Battuta is a joint Digital Government Initiative project being conducted by Iowa State University and the University of California, Santa Barbara, funded by the National Science Foundation, with matching support from other U.S. Federal and private sources. The project has been working on integrating heterogeneous geo-spatial information resources using flexible architectures for adaptive data collection in mobile environments, at the heart of which is a graphic user-machine interface. Software and systems architectures have been developed for several configurations of high-mobility computing devices, including palm tops and touch panel displays (Nusser et al, 2001). The UCSB component of the work has focused on wearable computing (Clarke, 2001).

Wearable computers represent a new generation of highly mobile computing devices. The user interface design, together with the placement of component devices on the body and their broader potential impact on society, are regarded as three major issues that will determine the future of wearable computing (Martin, 2002). There have been limited research on interface design for different applications (Althoff *et al*, 2001, Blaskó, 2002, Brewster *et al*, 2000). However, there is still no user interface specifically designed for the alternative interaction modes used on wearable computers for a geographically oriented system. A wearable GIS system requires the management of multi-source and multi-scale geographic data and their display to the user via a Graphic User Interface (GUI). In this GUI dynamic visualization and multi-modal interaction are indispensable, as also are the interactive use of data streams from the various spatially based input devices. In this paper we discuss the design and implementation of a prototype cartographic user interface for wearable computation that has been part of the work of project Battuta.

The remainder of the paper is organized as follows. Section 2 gives an overview of the hardware configuration of our wearable computer. Section 3 examines the management of high-volume geographic data with multi-resolution imagery, multi-scale map and dynamic field data. Section 4 discusses the implementation of dynamic visualization of geographic data. Section 5 reports on our user interface design and its means of multi-modal interaction. Finally section 6 gives the conclusions and discusses future work.

2. HARDWARE CONFIGURATION

The first generation UCSB wearable computer uses the CharmIT™ kit, built on the PC/104 specification, an industry standard for embedded computing for nearly ten years. The PC/104+ configuration evolved as an extension and includes a second 32 bit PCI-based bus. Our most recent version is now based on more advanced CharmIT technology and the regular PC computing standard, running at a clock speed of 800MHz.

Input and output devices were added to the core system. For text input, we use the Twiddler2, a one-handed keyer. With only 18 keys, the Twiddler2 keyboard is designed to perform all functions of a full, 101-key keyboard by "chording". A thumb-driven mouse is included. In addition, we have used a small keyboard shaped and strapped to the left forearm, and a small mouse worn something like a ring on one finger the left hand and moved with the thumb and by squeezing the hand. The primary output device is a MicroOptical SV-3 display. The company offers two color monocular (320x240) displays, a clip-on display and an integrated eyeglass display that magnifies the image displayed and, therefore, provides for a range of resolutions. The eyepiece of the clip-on display is suspended from a clear support and provides minimum occlusion by allowing a vision around all four sides of the virtual image. Field tests have shown that the clip-on monocular display is preferred, largely because it is far more flexible in use. While the system has proven effective in sunlight and is particularly good in low lighting conditions, it is not yet suitable for use in rain or high humidity.

Rechargeable lithium ion batteries have been the standard for wearable systems, and the Battuta system uses two larger camcorder batteries along with smaller batteries embedded in devices. Storage devices for wearable systems are magnetic hard drives, PCMCIA hard drives, flash memory, and external storage such as a USB hard drive. Our system used a primary hard drive contained in the same case as the CPU. The GPS unit is a Garmin 12-channel receiver mounted on the shoulder with velcro. To provide better orientation in space, the UCSB system uses a fluxgate compass with three-dimensional amplified sensing capability, and measures the users view roll, pitch and yaw continuously. This unit is shown on the shoulder in Figure 1, but can also be used on the wrist.

In summary, the UCSB wearable computer supports both text and GPS input, and augmented reality visual output. The system components are wired together through pockets and pouches contained in a simple vest. A photograph of the system in use is shown in Figure 1.

3. GEOGRAPHIC DATA MANAGEMENT

In a wearable computer system aimed at adaptive sampling in the field, multi-source and multi-scale geographic data are required to work in an integrated way. As part of the Battuta project, we have examined the conflation or geometric merging of data, and we have decided that the spatial data server will provide this service. While most field tasks can be accomplished with a detailed view of a smaller geographic space, such as a farm, the user still needs an overview scene for the larger area. This makes the efficient management of the multi-source and multi-scale data a necessity before the user interface can finally allow interaction.

Figure 1. The Project Battuta Wearable Computer

There are three major categories of data to be considered, map data, imagery data and field data. Map data and image data are static while field data are dynamic and incremental as fieldwork proceeds. In the wearable's database they fall into raster data and vector data. Static data act as geographic context and background for the dynamic data, which include GPS traces, sample points, etc.

In order to visualize multi-scale vector maps and multi-resolution images, and both dynamic and static data in an integrated way, images are placed into a multi-resolution image pyramid structure, while vector maps are organized into multiple layers according to their scale. This makes system performance totally independent of the extent of the geographic area in use. A geographic database can be then visualized according to the user's chosen scale for different field and indoor operation tasks.

3.1 Pyramid Image Structure

When taking the image of a large area into consideration, the pyramid image structure is the best approach to manage an image database. The image pyramid is capable of accommodating mass data volume and yet has high retrieval efficiency. It consists of a sequence of re-sampled copies of an original image in which the resolution is decreased in regular steps, often powers of two in size (Adelson *et al*, 1984). For different applications different re-sampling strategies can be used to make the resolutions changed both linearly or non-linearly.

For a large area, the image at each level could be immense in both dimension and storage, especially for lower levels of the structure. For efficiency of processing, the image is split into image tiles with the common shape of a rectangle. The size of the tile should be a compromise between load time for each tile and the number of tiles necessary to load, which is somewhat determined by the display size. For each display window tiles less than 2*2 should be best with a load time that is still acceptable.

An example of the tiled image in pyramid structure is shown in Fig. 2. For image pyramid construction, the user is required to specify three parameters, level number of the structure, tile size and size of the image on the top of the pyramid.

This structure allows the mass data storage for large geographic areas with some redundancy. The most important advantage of the structure is that the user can retrieve data for any sub-area of interest with both good resolution and high performance. From a zoom factor which determines how detailed the image is, and also how large the observed area is, a set of adjoined image tiles are extracted from the structure and merged into one image.

Figure 2. Pyramid structure with tiled images

3.2 Multi-scale Maps

As with the image, vector maps can be multiply scaled in the display used by the wearable computer. As the area of interested gets larger, the scale of a vector map should get smaller to make handling data more efficient and to keep the screen map legible.

A vector map differs from an image in that the map is composed of discrete spatial features such points, lines, polygons and text, and so the change of scale means the generalization and elimination of features. So the map at a higher level (smaller scale) cannot be derived directly from the lower level (larger scale) by automatic generalization, and so differently scaled maps need to be compiled beforehand. For a large area, it is also necessary to organize a map into sheet for conventional and computational reasons. Each sheet can be geometrically or geographically divided, as with the image pyramids.
We used an approach of spatial indices for the efficient storage and retrieval of multi-scaled maps, using a two-level index, one of which is the map sheet index and another is a map feature index. The map sheet index is constructed by dividing the whole area of use into grids and containing and intersecting map sheets for each grid are recorded, so that the required map sheet for any sub-area can be determined rapidly. The map feature index is constructed by dividing each map sheet into grids and the contained and intersected map features for each grid are recorded, from which the map features in the sub-area can be retrieved.

3.3 Field Data

Field data are intermediate and are collected data during fieldwork. For our application, they included GPS point and track data, digital compass data, stored navigation data and prior sample location data. For some of the planned applications, other thematic data such as land-use, soil type, etc. will be included.

Field data are application specific data, but part of each record contains when and where the data were collected. The location and the attribute data are recorded in real-time and on-the-spot and can be post-processed after the fieldwork or progressively analyzed either in the wearable or on a server connected via a network. These data are stored in the wearable computer in files separated from the geographic base data.

There are currently many data structures to accommodate the field data. A log is used to record the GPS data and digital compass readings, including geographic location, time, velocity and heading. These data are continuously tracked and written to disk. A track structure is used to record user-defined path information, including GPS and digital compass data for specific paths such as navigation routes. We include some pre-set start points and destinations. These tracks can be saved, loaded and replayed to repeat the field navigation.

4. DYNAMIC VISUALIZATION

A wearable device is a real-time and context-aware system. The user interface is driven by two kinds of real-time signals: a stream of location coordinates from a real-time GPS receiver chip, and roll, pitch and yaw inputs from the digital compass. These signals partially take the place of keyboard and mouse events. This dynamic multi-modal information should be instantly visualized in the user interface and should lead to an interactive process between human and machine. Such dynamic visualization and interaction is the most important component for a spatially-aware wearable system.

4.1 Software

The system software was developed using Swing, Java Advanced Imaging (JAI) components of the Sun Java 2 development platform. This allowed the support of image processing, gave a rich set of GUI widget components, and of course provides platform independence for the wearable and other mobile systems.

Swing is the newest Sun Java component set for the creation of graphical user interfaces with a cross-platform look and feel. Most conventional graphic user interfaces elements can be assembled based on the Swing components including menus, panels, and buttons, slide bars, etc. Swing was also found suitable as an API set for some of the non-conventional GUI elements required for the wearable computer because of its enhanced support for more flexible graphic widgets such as irregular buttons.

JAI is an especially important component for image processing. We found JAI to be cross-platform, device independent, highly interoperable, extensible and powerful enough to meet the requirements of professional-level image processing, including encoding and decoding different of image format, input and output, geometric transformations, rendering, color processing, image display and analysis. Nevertheless, the coding of the GUI required several hundred work-hours and even more for debugging and testing.

4.2 Visualization Approach

Dynamic cartographic visualization is still relatively new to digital cartography and GIS. The introduction of dynamic data adds another dimension, adding a temporal dimension to traditional visualization methods. Dynamic visualization is one of the defining characteristics of the wearable GIS system.

There are two reasons to support dynamic visualization. One is for the display of mobile or changing objects such as a GPS-equipped vehicle; another is for the display of the geographic environment relevant to a mobile subject, in our case a walker equipped with the wearable computer. The former needs a static geographic background and mobile objects where the viewer is also supposed to be static. This is called World Geographic Reference (WGR). In the latter, a moving geographic background passes a static object representing the moving user. This is called an Egocentric Geographic Reference (EGR).

Our system supports both WGR and EGR frameworks, partially because we seek to determine user preferences for different navigation tasks. For WGR, when the mobile object arrives at the edge of window, the current window is changed automatically to the adjoined map tile with some overlap as a buffer to keep the object always inside the display window. For EGR, the displayed map keeps moving backwards to reflect the mobile object which is fixed at the center of display window. Both WGR and EGR are useful for field and indoor applications, where WGR is more for the use of monitoring while EGR is for navigating.

As mentioned above, raster and vector data are stored in different data structures and have their own implicit or explicit geographic coordinate system respectively. To associate them into an integrated display an identical coordinate reference is necessary for exactly superposing display. The image pyramid that we use in our system takes GeoTIFF as its source and uses the UTM coordinate system. Within each GeoTIFF image, an affine transformation is used to change the UTM coordinate to pixel-based coordinate of image (Clarke, 1995). Vector data are required to use the

UTM coordinate system too, which are in turn associated with image data by the identical user coordinate system – UTM (Figure 3).

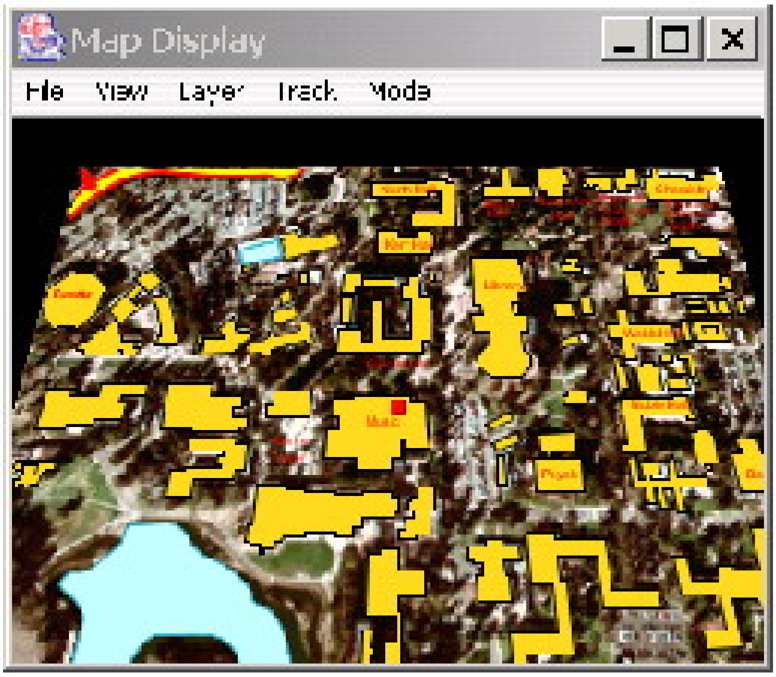


Figure 3. Superimposing an image on a map

For different applications, and to facilitate usability studies, the system is designed to have alternative visualizations for the image and map data, and to allow the user choices. There are different factors that control how the geographic data are visualized upon the interface, which falls into two categories: dynamic signals, and user-controlled degrees of freedom. These are characterized as follows.

❿*Dynamic signal*: Signals from the GPS and digital compass are primary controls of the appearance of the visualization, where moving of map or object is driven by GPS and rotation of the map by the digital compass. Recorded dynamic data can in turn replay the visualization after the fact.

❿*User controllable DOFs:* In our GUI, the user has control over the degrees of freedom in the display, which include:

- *Zoom factor:* A user can choose how far the viewpoint is from the earth's surface, thus determining how large the visible area is and how detailed the representation is.
- *Orientation:* the user can choose which direction is upward, thus view the scene from different directions.
- *Image/map toggle*: A user can choose whether an image, a map or both are visible.
- *Perspective/orthographic view:* A user can toggle between perspective and orthographic viewing modes. The user can also specify how high the viewpoint is above the earth surface in the case of the perspective mode.
- *Egocentric/world reference:* A user can toggle between EGR and WGR frameworks.
- *Real-time/interactive mode:* A user can also toggle between real-time and interactive modes. The former is for fieldwork and navigation, where some key DOFs are controlled by real-time signals such as location and orientation. The latter is for indoor use and post processing, where all DOFs can be controlled by the user via a mouse or keyboard, when the user can replay, edit and analyze the field data.

All these DOFs can be use in combination. For example, the user can choose WGR while keeping orientation and with the zoom factor changing. Multiple DOFs and their combination provide a user with great flexibility and make the wearable system suitable for a large array of tasks in field navigation, planning and decision-making.

4.3 Text Labeling

One of the particularities in the wearable system is that we introduce 2.5-dimensional representation into the dynamic visualization. We are exploring how to best represent the map efficiently in such a dynamic and non-orthographic situation We find that text labeling is especially important, because the text must stay well placed and legible while other map features are moving or rotated, a fact complicated by the aligned menus. Text is different from other objects in the map in that it must remain vertical for legibility. Text must be separated from other map features and visualized using upward alignment (Figure 4). Map rotation can mean that multiple text strings that belong to one geographic feature (e.g. a building) should be handled as one so that they will not become misaligned.

Figure 4. Text labeling in a dynamic map

4.4 System Architecture

Java is an Object-Oriented language. With Sun Java 2, our wearable system consists of a set of modules which are Java classes:

- *Main module*: a self-running thread program to coordinate the operation of the device module and the map display module. It continuously requests the GPS and digital compass signals and send them to the map display module to adjust the visualization.
- *Map display module*: another self-running thread program to listen for and dispatch various messages from the various inputs mouse, gesture input, keyboard and etc. Gesture control is a new option, and involves a lipstick camera mounted on the users eyepiece, and pointed at the users hands. Third party software interprets the gestures using image processing, and returns messages. In the map display module most DOFs are implemented to control the appearance of visualization. Classes are implemented as follows:
- *Pyramid image class:* gets image data from the pyramid structure, transforms and displays them according to preset DOFs.
- *Multi-scale map class:* gets map data from the cartographic database, transforms and displays them according to preset DOFs.
- *Geographic object class:* handles geographic objects in the cartographic database.
- *Spatial index class:* handles the spatial index system in the cartographic database
- *Geo-transformation class:* a set of image and map transformation methods such as rotation, scaling, translation and perspective warping.
- *Dynamic Data Module:* handles derived data from input device signals.
- *GPS object class:* handles current location of moving object.
- *Tracking class:* handles tracking information the moving object passes.
- *Device Module:* handle hardware signals via an interface to serial ports on the computer and transforms them into a usable format.
- *GPS class:* monitor the USB interface and input geographic coordinates, time, velocity and direction etc. and transform geographic coordinates to UTM.
- *Digital compass class:* monitors the USB interface to receive the heading direction of a moving object.

5. GUI-DESIGN AND MULTI-MODAL INTERACTION

The GUI for the wearable system is the most critical component for the success of the entire approach. If it is too complex, the system will not be used at all. If it is not well designed, the system will not be used effectively, or indeed better than current solutions. If the GUI is effective, it will allow the user full control over the multiple modes of input that the system supports**.**

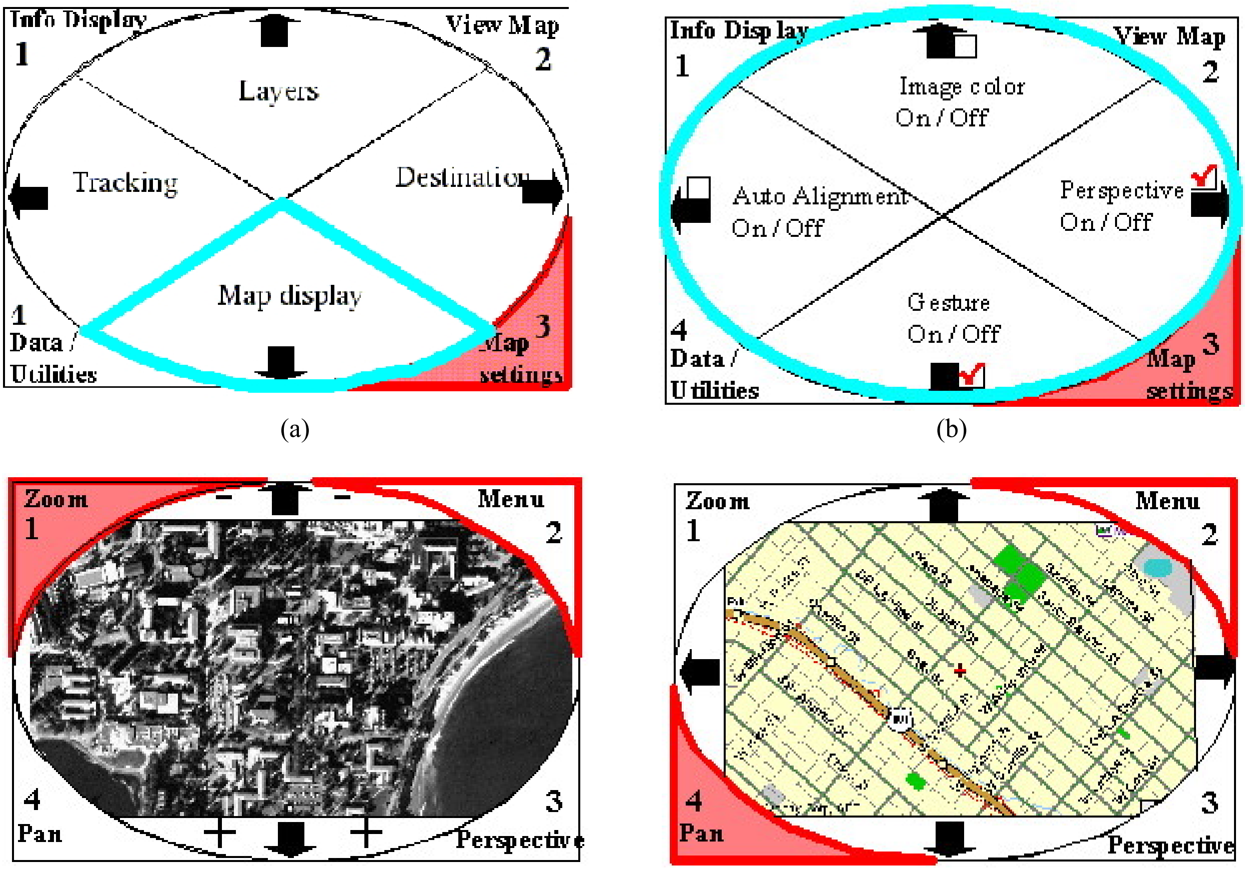
5.1 Icon-based Interface and Menu System

In an attempt to depart from the windows, icons, menus, and pointers (WIMP) approach, the use of pointing devices has been minimized so that the user interface can be controlled predominantly with keyboard commands to work toggle switches and selection mechanisms. Sounds has been explored and rejected as an alternative input and control mechanism.

Figure 5 (a) shows the four main menu choices in the four corners of the display. These can be selected by entering numbers one through four (or any other more intuitive key combination) on the keyboard, or by mouse clicks. Subsequently, four corresponding submenus are displayed in the center-viewing oval (Figure 5 (b)). Any of these four choices can be accessed by entering the matching up, down, right, or left arrow key. The choices at the bottom of the decision tree can be selected and de-selected by hitting the appropriate arrow key again. Figure 5 (c) shows how, after selecting "*View Map*" in the upper right hand corner, this menu item changes to "*Menu*", giving the user the chance to return to the main menu. At the same time, a map is display in the center oval, and the other three menu items in the corners change to map specific functions. If "*Zoom*" is selected (Figure 5 (c)), two menu bars afford zooming in and out by pressing the corresponding arrow key. If "*Pan*" is selected (Figure 5 (d)) four menu bars afford panning in the four principal directions by pressing the corresponding arrow key. All menu choices here are also accessible with a pointing device, such as the single hand mouse.

5.2 Multi-modal Interaction

We have designed a series of unobtrusive interface driving mechanisms for the GPS, the digital compass, a one-handed keyboard, a thumb-driven mouse, gesture input and also virtual input. These inputs allow the user interface be handled in a very flexible way both in field and indoor operation, and are permitting the Battuta team to conduct a set of field experiments aimed at refining the user interface. Building input signal simulators was a critical first step in the software design before actual components could be built and tested. The gesture input exploits the modular nature of the interface code, and requires only minimal interaction with the main modules. This work is being done in collaboration with the Department of Computer Science at UCSB.



(a) (b) (c) (d)

Figure 5. Parts of the graphic user interface

6. DISCUSSION AND FUTURE WORK

The user interface for spatially aware wearable computation has all of the common characteristics of desktop GIS and yet adds others, among which dynamic visualization and multi-model interaction are the most important. In this paper we have taken mass image and map data management into consideration in designing a prototype cartographic user interface which can meet basic requirements of field adaptive sampling. It is concluded that mass geographic data can be incorporated with dynamic visualization techniques and multi-model interaction between human and machine to develop better wearable and context-aware system. In the next round of our work, we plan extensive field tests to observe and record actions taken by users in simple navigation tasks.

Apart from incorporating dynamic visualization and multi-modal interaction into the GUI, we have introduced visualization strategies which are crucial for a real-time geo-computation system, including perspective visualization and multiple DOFs. We still foresee the possibility to introduce others that more closely conform with human perception and are more context-aware, including three dimension visualization and Virtual Reality technology. Nevertheless, the research is rapidly leading the way toward a practical, advanced, highly mobile cartographic system that may be of great practical use in a plethora of human applications. Given that wearable computers are both falling in price and becoming more commonplace, the device independency of our software approach means that it will be compatible with most future systems.

7. ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation of the United States under the Digital Government Initiative, and by the project partners. Further information about the project is available at: http://dg.statlab.iastate.edu/dg/. We are grateful to Wuhan University, for allowing Dr. Qingyun Du to spend a sabbatical year at the NCGIA in Santa Barbara to work on the project.

8. REFERENCES

[1] Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M. (1984) Pyramid Methods in Image Processing, *RCA Engineer*, 29(6):33-41, 1984

[2] Althoff F., Volk T., McGlaun G., and Lang M. (2001) A generic user interface framework for VR applications. *9th International Conference on HCI*, New Orleans, August 2001.

[3] Blaskó, G. (2002) A Menu Interface for Wearable Computing, *Proceeding of 6th IEEE International Symposium on Wearable Computers,* Seattle, WA, 7-10 October, 2002, p164-165

[4] Brewster, S.A. and Walker, V.A. (2000). Non-Visual Interfaces for Wearable Computers. *IEE Workshop on wearable Computing* (00/145). IEE Press.

[5] Clarke, K. C. (1995) *Analytical and Computer Cartography*. 2 ed. Upper Saddle River, NJ: Prentice Hall.

[6] Clarke, K. C. (2001) Cartography in a mobile internet age. *Proceedings of the 20th International Cartographic Conference*, ICC 2001, Beijing, China, August 6-10, 2001, Vol 3, 1481-1488.

[7] Martin, T. L.(2002) Time and Time Again: Parallels in the Development of the Watch and the Wearable Computer, *Proceeding of 6th IEEE International Symposium on Wearable Computers*, 7-10 October 2002 • Seattle, Washington

[8] Nusser, S., L. (2001) Miller, K. Clarke, M. Goodchild. 2001. Future views of field data collection in statistical surveys. *Proceedings of 2001 National Conference on Digital Government Research*, Los Angeles, CA.

A PROTOTYPE CARTOGRAPHIC USER INTERFACE FOR WEARABLE COMPUTING

Clarke, K.C.[1], Du, Q.[2], Nuernberger, A.[1] and Pingel, T.[1]

[1]Department of Geography-NCGIA.University of California
Santa Barbara, CA 93106-4060, USA.
[2]School of Resource and Environmental Science, Wuhan University,
430072 Wuhan. PR China.

Biographies

Dr. Keith C. Clarke is a research cartographer, professor and Chair of Geography at the University of California, Santa Barbara. He holds the M.A. and Ph. D. from the University of Michigan, specializing in Analytical Cartography. His recent research has been on modeling urban growth using cellular automata, on mobile computing, and on the history of the CORONA remote sensing program. Dr. Clarke is the former North American Editor of the International Journal of Geographical Information Systems, and is series editor for the Prentice Hall Series in Geographic Information Science. He is the author or co-author of three textbooks and about eighty book chapters, journal articles, and papers in the fields of cartography, remote sensing, and geographic information systems. Dr. Clarke has been a NASA /American Society for Engineering Education Fellow at Stanford University, Science Advisor to the Office of Research, National Mapping Division of the U.S. Geological Survey, is currently the Santa Barbara Director of the National Center for Geographic Information and Analysis, and chaired a recent committee for the U.S. National Academy of Sciences.

Dr. Qingyun Du is Professor of Cartography and GIS, and deputy director of the School of Resource and Environmental Science at Wuhan University. His research interests include linguistic conceptual models of geographical information, automatic cartographic generalization, electronic mapping and spatial visualization. He serves on 2 editorial boards for journals. He is corresponding member of ICA Commission on Theoretical Cartography and co-chairs for two Chinese association commissions on GIS and Cartography. He has led more than 30 projects and received 8 national and ministerial scientific awards.

Andrea Nuernberger is a graduate student in the Department of Geography and NCGIA at the University of California, Santa Barbara. She has interests in mobile computing and behavioral geography, and has worked on the Battuta project since its inception.

Tom Pingel is a graduate student in the Department of Geography at the University of California, Santa Barbara. He has interests in digital mapping in cyberspace, computer security, and mobile applications. He is responsible with Andrea Nuernberger and Dr. Du for most of the GEORGE user interface design and Java coding.