

MIKAN GLOBE MT: An Interactive World Map featuring Seamless Map Projection and Guaranteed Visibility

Katsutsugu Matsuyama Makoto Okamoto

Future University-Hakodate

{kmatsu, maq} @ fun.ac.jp

Abstract

This paper describes the research and development of a new tool for visualizing world maps. The developers anticipate this tool will provide a better grasp of structures and enable an improved understanding of features from multiple viewpoints. The following needs were identified on the basis of interviews with subjects who are likely users of map software: overhead views and true-to-life views; side-by-side comparison of arbitrarily selected locations; and context awareness. The software was designed to satisfy these requirements by incorporating a new algorithm, resulting in a package with distinctly different characteristics from those that are browsed in a directly geometric fashion. In this paper, we propose a viewing system that allows the user to switch seamlessly between different geographical projections and a multi-focus view displaying method by parameter adjustment and split screen. The effectiveness of this approach is validated in a survey of subjects using this mapping software.

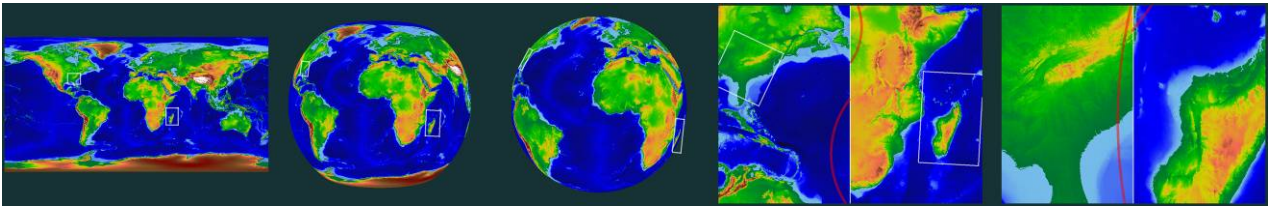


Figure 1. Screen-shots of MIKAN GLOBE MT. The views change seamlessly while zooming in. Display parameters and split-screen allocation are automatically adjusted to keep the location selected by the user within the display.

1. Introduction

Applications of interactive world maps are one of the most commonly used types of software. Most of the software packages in current use show geographical features mapped directly onto the surface of a sphere or flat surface. To date, most studies of user interactions with world maps have only considered 2D maps [1, 2]. Most three-dimensional (3D) interactive mapping techniques display the spherical primitives directly on the user's monitor [3, 4] or employ spherical displays [5, 6], which implies that data visualization with virtual globe applications has not received much attention.

In preparation for the development of this software, world map researchers, educators and ordinary users were interviewed for their views regarding the current situation in mapping software and desirable functions for upgraded versions of such software. The interview subjects were chosen on the basis of personal introductions. The following capabilities were regarded as the most difficult to obtain from current applications, among the functions desired by the subjects:

Cartography researchers: There are locations that are physically distant from each other but which share similar climates and topographies. New discoveries are expected to result from the ability to simultaneously display and observe such locations.

Cartography teachers: We have been teaching projection methods with static aids, drawings in textbooks and on the blackboard. It will be very helpful to have a tool that shows these methods directly.

Researchers on field servers and satellite data-transmission systems: Data is usually pasted into Google Earth because it is easy and fashionable. We need an application that allows us to manipulate measured and gathered data more effectively and to examine it from many different viewpoints.

1.1 Requirements

We examine the capabilities required of new map software, as revealed in the responses to a survey.

Overhead views and true-to-life views: users need both overhead views, which provide a macroscopic panorama of the distributions and trends of the data over a wide area, and globe-style views that minimize the distortions inherent in

cartographic projections. Since virtual globe applications are not capable of showing two locations on opposite sides of the globe, the user must move the camera or rotate the globe to see one or the other. World map software provides an overhead view, but the projection method is not very accurate, as it introduces distortion.

Multi-focus: The software must be capable of measuring, observing and surveying different locations simultaneously to carry out comparisons between them. The results of our interviews revealed that there is a widespread desire for the capability to make comparisons at fine levels of resolution of any grouping of arbitrarily specified locations.

Context awareness: World map applications handle data covering the entire globe, which is an immense volume of data. It is sometimes difficult to perceive existing structures if discontinuous changeovers occur during operation.

1.2 Contributions

This paper proposes a framework that satisfies the requirements given in the previous section. To date, most of the studies of interactions of users with world maps have only considered 2D maps [1, 2]. Most of the 3D interactive globe techniques directly display the spherical primitives on the monitor of the user [3, 4] or employ spherical displays [5, 6]. However, neither of these requirements fulfills the requirements stated in this paper. One of the special features of this software package is that it can handle geographical data in both 2D and 3D, providing the user an interactive environment with a high degree of freedom (DOF), i.e., a wide variety of options. The specific contributions of this paper are as follows.

A viewing system that allows the user to switch between cartographic projections: This software provides a viewer that combines 3D views, as normally viewed on a globe, with 2D views, as seen on an ordinary map, and allows seamless switching between the two views. The view can be toggled automatically or manually according to the aims of the user.

Guaranteed visibility: The above features of the viewing system guarantee that any region designated by the user can be kept within the borders of the display. There have been some papers on guaranteed visibility for large volumes of tree-structured data [7] and for 2D data such as images [1, 2],

but none of those algorithms can be used in high-DOF viewing software allowing switching between projections.

This approach of using high-DOF viewing software and the ability to switch between views will facilitate the examination of pre-existing fixed maps and consequently the discovery of new information that had previously been "buried" in the maps. This software could also be used in tools designed to aid user understanding of map projections and for creating new maps. In addition to representing spatial information, such as GIS and cartographical data, this approach can also be used in games, art and a wide variety of other applications.

2. Related Research

This section examines some existing techniques for their similarities to our method.

Pan & Zoom: Pan and zoom are the basic operations needed when the visual space is larger than the viewport. Papers have assessed how to combine Pan and Zoom in terms of space-scale diagrams [8] and camera work trajectory designed to prevent stress to the viewer [9]. Other papers have analyzed procedures for maintaining a constant scroll rate by automatic and seamless zoom-out, as a function of the scroll distance [10] as well as high-speed procedures for Pan and Zoom with a single DOF of scroll using the rectangular mouse coordinate system [11].

However, most of the above research involved content with two dimensions, or even a single dimension, and is not directly applicable to this high-DOF algorithm.

Focus+context using distortion: Another approach is to deliberately distort the space in a non-linear fashion instead of searching for a particular location using Pan and Zoom. A common example is the use of a fish-eye view [12, 13]. One method for doing this with 2D spaces can be metaphorically compared to "stretching" a rubber sheet [14]. The "rubber-sheet approach" directly distorts the contents of the map. This principle can also be indirectly applied to the frames used to reduce the visual distraction of this approach [7]. Other papers have suggested procedures to "fold up" the space in the depth direction [1] and to employ semantic zooming to introduce distortion [15].

Distortion procedures appeared to be good candidates for replacing Pan and Zoom. However, maps already contain some distortion due to projection and there are legitimate concerns about confusing the user with further distortions, as well as concerns about complicating mouse operations by combining distortion adjustment with the already available Pan and Zoom operations. As a result, we decided not to incorporate distortion procedures.

Split screen: Split screens are already widely used. The overview window provides a view of the current target in an overhead view of all of the contents of the map, regardless of the

contents of the other windows, similar to the presentation style of Adobe PhotoShop. Since the overview window cannot show all the details, it has disadvantages from the viewpoint of context sensitivity. However, note that the overview window is not an exclusive approach. A procedure has been proposed for automatic splitting of the display into separate windows in response to certain interactions in 2D spaces [2]. This paper uses a modified version of this split window method while providing a high DOF.

Cognition of off-screen targets by icons or other symbols: Several stratagems have been proposed to help cognize off-screen targets. The *Halo* is a circle centered on targets [16]. Its purpose is to cognize the distance and azimuth toward the target. *City lights* shining from the targets cast shadows on the edges of the display [17]. Other approaches replace off-screen targets with rectangles [18] or designate a substitute object for an off-screen target and pan the view automatically so that the target is displayed on the monitor [19]. Halos were selected for incorporation in this algorithm to help cognize targets between different windows when the user is using a divided screen.

Panorama techniques: Globe applications show a sphere viewed from the outside. Conversely, the surface can also be shown from the inside looking out: the provision of techniques for showing a panorama of the environment from an interior point was the motivation for developing this algorithm. Kopf *et al.* developed a procedure for capturing a gigapixel image and an effective viewer for presenting it [20]. They solved the problem of the distortion that occurs in previous viewers when the angular field of view is too wide by bending the projection screen in response to the zoom selected by the user, resulting in a seamless display of an overview during zoom-out. This is a new method for toggling continuously between overhead views and detailed views in a broad visual space. This research was inspired by this panorama viewer and we have adapted the concept of seamless bending to a spherical shape.

Interactive techniques: There are several interactive techniques adapted for use with a globe. Goetz *et al.* have proposed a high-resolution texture mapping procedure for 3D interactive globes [3]. This procedure assumes that the object is spherical, and is thus not applicable for use with the present method which employs high DOF manipulations of an interactive world map. Ueda has proposed a model that divides the surface of the interactive 3D globe into a mesh and conducts calculation using industrial structures as geographical parameters [4]. Companje *et al.* have developed an installation that projects data about the Earth's surface onto an acrylic sphere [5]. A track ball in contact with the sphere facilitates rotation of the sphere; rotating a disk attached to the side of the sphere allows the user to vary time parameters, change the sunlight conditions, and to observe continental drift. Benko *et al.* have

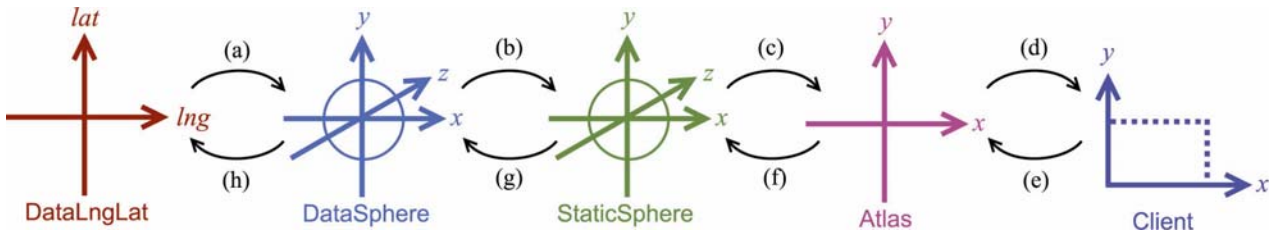


Figure 2. Series of coordinate systems used in this method.

proposed a global display that permits multi-touch interactions [6]. The user can perform a great variety of operations including rotation of the sphere by tracing across the display surface with their fingers.

3. System for Toggling between Projection Schemes

This section proposes a viewing system that combines the 3D view of a globe with the 2D view of an ordinary map [21]. Switching between the globe-viewing mode and the map-viewing mode can be done manually or automatically, depending on the aims of the user. The capability for continual switching is essential because it enables the user to get a good grasp of structures. Rather than using the standard camera motions commonly found in 3D computer graphics, this algorithm features operation of parameters for transformation functions between multiple coordinate systems. It expresses switching between views as deformations of a sphere: this approach is necessary to secure the axes employed in defining the deformations.

Maps are made based on one of two available projection methods. This algorithm employs equidistant cylindrical projection and azimuthal equidistant projection. Equidistant cylindrical projection uses orthogonal gridlines, with verticals for longitude lines and horizontal for latitude lines: it provides true distances on the equator and on longitudinal lines. In azimuthal equidistant projection, the map is drawn along polar coordinates from a central point: its advantage is that the distance and direction from the central point to any other point is correct [22].

This method uses the series of coordinate systems shown in Figure 2. It is composed of five coordinate systems with transformations established for each pair of neighboring systems (Fig. 2 (a)-(h)). The geographical data is maintained in the 2D DataLngLat system and ultimately displayed on the user's monitor in the client system. The purpose of this section is to determine the functions for the transforms (a)-(h).

The role of each coordinate system is as follows.

- DataLngLat (2D): Geographical data is stored in this system. This corresponds to the texture coordinate system in computer graphics.
- DataSphere (3D): This corresponds to the model

coordinate system and expresses the Earth as a sphere with a radius of 1, centered at the origin. The y-axis expresses North.

- StaticSphere (3D): This corresponds to the world coordinate system, a rotated version of the DataSphere globe. Therefore, in this system, the y-axis does not necessarily indicate North.
- Atlas (2D): This is like the screen coordinate system, the projection of the StaticSphere system onto a screen.
- Client (2D): This is the coordinate system that is displayed on the user's monitor. The origin is at the upper left corner of the screen and the display size is the viewable area of the User's monitor (dotted lines in Fig. 2).

Each coordinate system resembles one or another of the coordinate systems used by standard computer graphics, but a key feature of this approach is that the parameters of each transformation function depend on the parameters of the other transformation functions.

3.1 Coordinate Transformation Functions

The transformation functions for the coordinate systems shown in Fig. 2 are as follows.

Transformation (a): The following equations transform the spherical coordinate system, using latitude and longitude on a sphere with a radius of 1:

$$\begin{aligned} x &= \cos(lat) \cdot \sin(lng) \\ y &= \sin(lat) \\ z &= -\cos(lat) \cdot \cos(lng). \end{aligned} \quad (1)$$

Transformation (b): The sphere is rotated as defined by rotation matrix \mathbf{M}_b , which is the product of three rotation matrices, $\mathbf{M}_b = \mathbf{M}_I \mathbf{M}_U \mathbf{M}_A$. \mathbf{M}_I expresses the rotation of the initial position and \mathbf{M}_U is defined by the operations of the user. \mathbf{M}_A is explained in Section 3.3, *Transformation of pan operations to rotation*; here it is set to \mathbf{I} .

Transformation (c): Transformation (c) consists of deformation of the sphere in StaticSphere for toggling between views and the screen projection.

Screen projection is performed by placing a screen and camera on the StaticSphere coordinate system. The screen is located on the plane of $z = -1$ and the camera is placed on the z-axis. In this procedure, the camera is at $(0, 0, -5)$. Thus, the screen and camera locations do not shift during deformation of

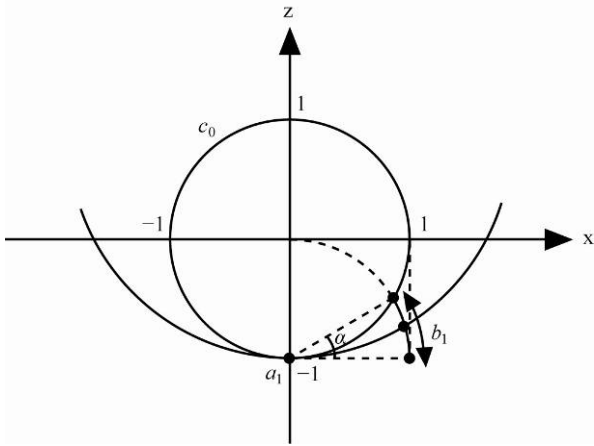


Figure 3. Deformations caused by parameter α .

the sphere. A perspective drawing is made of the sphere centered on the origin in the Atlas space, as was done in the globe view without deformation. The entire surface of the globe is projected, so if a massive volume of geographical data is used, the results are presented as a “high-resolution” map in the Atlas coordinate system.

Toggleing with the map view is done by deforming the globe. Deformations are performed in this procedure by the method of Kopf *et al.* with a projection screen [20].

A cross-section of the sphere parallel to the xz plane is employed for this explanation (Fig. 3). A circle centered on the z -axis is drawn through points $a_1 = (0, -1)$ and $b_1 = (\cos(\alpha), \sin(\alpha)-1)$ using parameter $\alpha = [0, \sin^{-1}(0.5)]$. An arc of that circle of length 2π is then deformed. When parameter $\alpha = \sin^{-1}(0.5)$, the arc becomes a circle of radius 1; the circle “opens” to an arc at lower values. In the special case of $\alpha = 0$, the arc becomes a line along $z = -1$.

If the same deformation is performed about the y -axis with parameter β , the final result is a segment of a torus, and when $\alpha = \beta = 0$, the same result is obtained as with an equidistant cylindrical projection [20]. The azimuthal equidistant projection can be obtained with a single deformation parameter in a polar

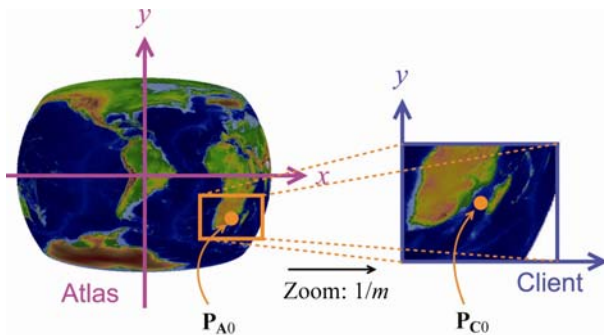


Figure 4. Transformation (d). The display is centered on \mathbf{P}_{C0} , \mathbf{P}_{C0} is the projection of \mathbf{P}_{A0} in the client coordinate system, and m is the expansion factor. \mathbf{P}_{A0} is determined by the user panning, and m is determined by the user zooming.

coordinate system.

Transformation (d): This transformation is performed to clip the target region from the high-resolution projection in the Atlas coordinate system. Specifically, the position vector \mathbf{P}_A in the Atlas coordinate system is transformed to \mathbf{P}_C in the client coordinate system by

$$\mathbf{P}_C = \mathbf{P}_{C0} + (\mathbf{P}_A - \mathbf{P}_{A0})/m \quad (2)$$

where \mathbf{P}_{A0} is the baseline position vector in the Atlas coordinate system: \mathbf{P}_{C0} is the projection of \mathbf{P}_{A0} in the client coordinate system, and therefore the baseline position vector there; and m is the coefficient of scaling (Fig. 4). The greater m is, the greater the extent of zoom-out will be. \mathbf{P}_{C0} is in the center of the display in this algorithm. Panning and zooming by the user change \mathbf{P}_{A0} and m respectively.

Transformation (e): Transformation (e) is the inverse of transformation (d). \mathbf{P}_A is unknown and \mathbf{P}_C is known:

$$\mathbf{P}_A = \mathbf{P}_{A0} + (\mathbf{P}_C - \mathbf{P}_{C0}) \cdot m \quad (3)$$

Transformation (f): The inverse of transformation (c), it is found by calculating the locations of intersections between rays from the camera location and the torus.

Transformation (g): The inverse of transformation (b), it is \mathbf{M}_b^{-1} .

Transformation (h): The inverse of transformation (a), it is found by solving Eq. (1) for lng and lat .

Table 1 shows the parameters of this series of coordinate systems and transformation functions that can be manipulated by the user. These parameters are fixed as necessary according to the aims of the user. For example, in applications that only use the globe view, the deformation parameter α is set to $\sin^{-1}(0.5)$, while in applications such as Google Map, which use map view, α is set to 0 and the rotation matrix $\mathbf{M}_U = \mathbf{I}$. Figure 5 shows an example where $\alpha = 0$ and the user has operated a rotation.

Auto-toggleing between views is done by making the deformation parameter α dependent on the coefficient of scaling

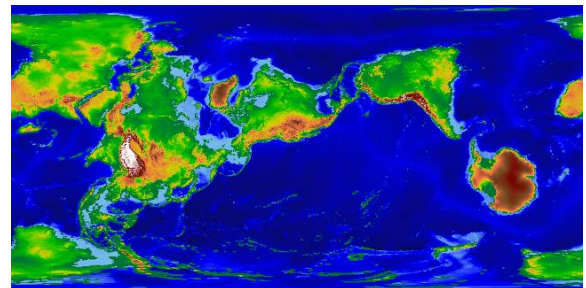


Figure 5. Example of the result of applying rotation in transformation (b) while $\alpha = 0$.

Table 1. User-set parameters.

Parameters	Transformation	Comments
\mathbf{M}_U	(b), (g)	Rotation
α	(c), (f)	Deformation
\mathbf{P}_{A0}	(d), (e)	Pan
m	(d), (e)	Zoom

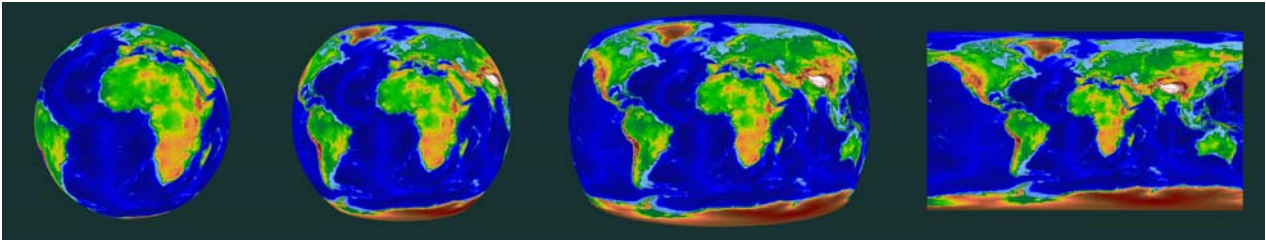


Figure 6. Example of switch-over to equidistant projection.

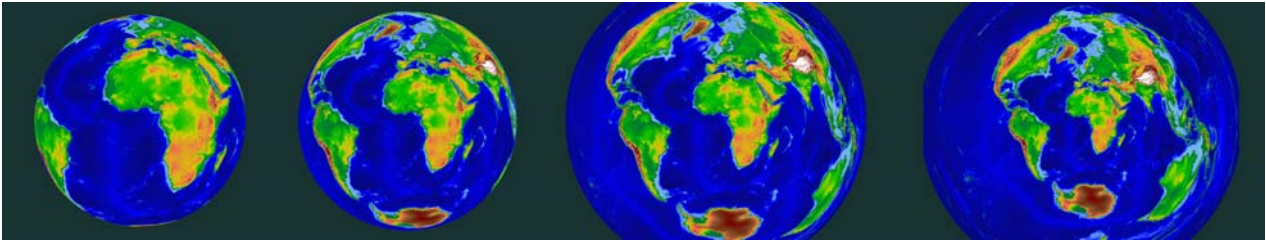


Figure 7. Example of switch-over to azimuthal equidistant projection.

m . This relationship can be set as required for any specific purpose, but in this paper, it is set with the following sigmoid function.

$$\alpha = \sin^{-1}(0.5)/(1+\exp(S_0 m - S_1)) \quad (4)$$

The coefficients are $S_0 = 2700$, $S_1 = 0.0047$. This displays the map view during zoom-out: the display is changed to globe view during zoom-in. Figures 6 and 7 show examples of deformation using equidistant cylindrical projection and azimuthal equidistant projection.

3.2 Assignment of Mouse Button Functions

Of the various transformation function parameters, the user can manipulate \mathbf{M}_U , \mathbf{P}_{A0} and m in rotate, pan and zoom (Table 1). Both rotation and panning involve motions of the viewpoint or the object, so it feels natural to perform these by dragging the

mouse. This brings up the question of how to distinguish between rotation and panning. We felt that it was unacceptable to require the user to press different mouse buttons for these functions, which are quite similar to each other, as they both involve motion. Furthermore, that approach would require the user to have a certain level of understanding of the internal working of the software. In addition, it would be necessary to assign the function of selecting regions to the mouse button.

Users tend to use pan to manipulate maps, and much less often to manipulate globe displays. In this algorithm, a threshold value μ is employed to place Rotate and Pan on the same mouse button: when a map is being displayed, i.e., when $\alpha \leq \mu$, and the button is operated, the map is panned; otherwise, operation of the button causes rotation. The threshold value was chosen by trial and error, and $\mu = 0$ was found to provide the most natural

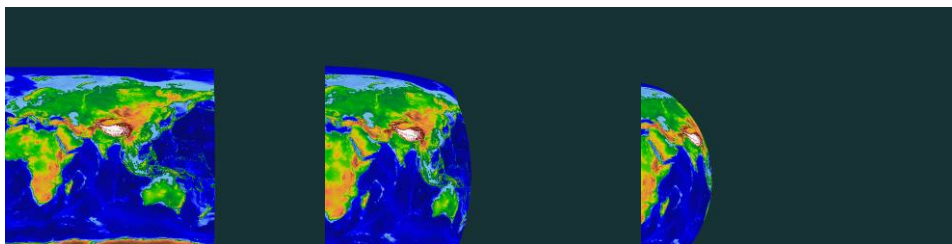


Figure 8. Problems occurring as a result of changes to deformation parameter α . If α is changed after a pan, \mathbf{P}_{C0} is displayed in a different location and the center of the globe is shifted to a different location than \mathbf{P}_{C0} .

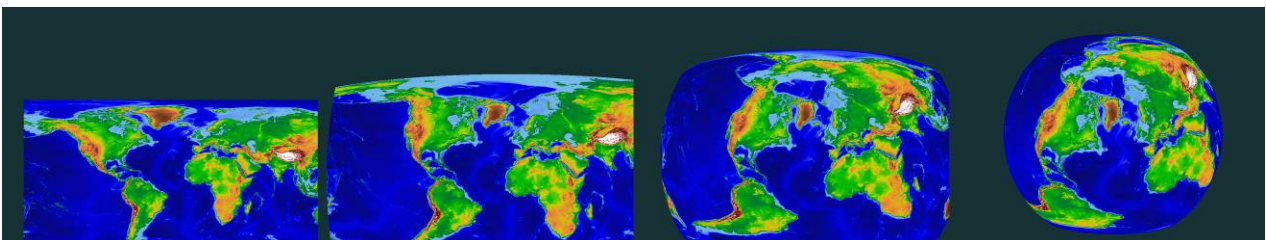


Figure 9. Example of switch from pan to rotate.

feel.

3.3 Transformation of Pan Operations to Rotation

Pan operations shift location \mathbf{P}_{A0} in the Atlas coordinate system, which corresponds to the center of the display (\mathbf{P}_{C0} in the client coordinate system), as shown in Eq. (2), (3) and Table 1. Here, if \mathbf{P}_{A0} is any point other than (0, 0) and parameter α is reset to change a projection (deformation of the globe), the display shown in Fig. 8 is created.

- Problem (1): The deformation of the globe in transformation (c) is performed about the z-axis in StaticSphere, so as α is changed, the data whose locations are determined relative to \mathbf{P}_{C0} shift.
- Problem (2): Since \mathbf{P}_{A0} is not (0, 0), the center of the globe is shifted to some location other than \mathbf{P}_{C0} , the center of the screen.

The changing of a projection method is not a motion in the sense that pan and rotation are motions, so if the displayed locations shift during a switch of projections, it will tend to confuse the viewer. Furthermore, when using globe view, viewers instinctively place the center of the globe at the center of the display.

Therefore, a process must be introduced that ensures that data locations defined by \mathbf{P}_{C0} are held stationary during changes to α , and that the center of the globe in a globe view remains in the center of the view.

If compensation is incorporated into this algorithm to ensure that the closer α is to $\sin^{-1}(0.5)$ the closer \mathbf{P}_{A0} is to (0, 0), it will resolve Problem (2). \mathbf{P}_{A0}' , the compensated location of \mathbf{P}_{A0} , is determined as follows.

$$\begin{aligned} \mathbf{P}_{A0}' &= \mathbf{P}_{A0} \cdot k \\ k &= 1 - \alpha / \sin^{-1}(0.5) \end{aligned} \quad (5)$$

Coefficient k shifts from 1 to 0 with the deformation from map ($\alpha = 0$) to sphere ($\alpha = \sin^{-1}(0.5)$). Parameter r determines the rate of compensation; in this algorithm, it is set at $r = 3$.

Correction for Problem (1) requires a rotation matrix that compensates for the difference between the data coordinate mapped to the compensated \mathbf{P}_{A0}' and the data coordinate mapped on \mathbf{P}_{C0} when $\alpha = 0$. This correction is expressed by rotation matrix \mathbf{M}_A for Transformation (b).

Rotation matrix \mathbf{M}_A can be set to various values, depending on the map projection method used. When equidistant cylindrical projection is used, $\mathbf{M}_A = \mathbf{M}_{Ax} \cdot \mathbf{M}_{Ay}$, where \mathbf{M}_{Ax} is a matrix to rotate the position vector about the x axis by the discrepancy in latitude and \mathbf{M}_{Ay} is a matrix turning the position vector about the y axis by the discrepancy in longitude. If azimuthal equidistant projection is used, \mathbf{M}_A must be a rotation matrix that creates the shortest possible path in the DataSphere coordinate system. Figure 9 shows an example of pan motion

being replaced with a rotate action.

3.4 Results of Switching between Projection Methods

Figures 1, 6 and 7 show examples of transforms using equidistant cylindrical projections and azimuthal equidistant projections. It can be seen that both of these methods provide smooth transitions from the global view. In addition to the ease with which these projections can be implemented, there were several reasons for selecting these two projections from the various map projection methods that have been described to date available. First, directions can be easily interpreted when using equidistant cylindrical projections: these projections produce world maps with outlines that resemble those that are commonly used to produce everyday maps and can therefore immediately be understood by users as maps representing overhead views of the globe. Azimuthal equidistant projections provide good functionality, since distances and angles from the center point are accurate. A great number of projection methods are currently available to the cartographer [22], and the choice of the method has a significant impact on the overhead view provided to the viewer. The authors believe that the viewer would benefit from the additional data conveyed to users by using additional projection methods and also feel it would be significant from the point of view of cartography educators. Our future tasks therefore include the implementation of additional projection methods and evaluations in the area of education and geographic studies.

Figure 9 shows an example of how Pan can be replaced by Rotate. The 2D pan in the Map view is seamlessly shifted into a 3D rotation. This replacement can be described as the placement of the object in its optimal orientation instead of a camera movement. It allows effective display of information besides geometrical shape data when there is a large amount of data attached to the view.

The manipulation modes separating Pan and Rotate can be implemented with relative ease and can be operated relatively smoothly. These functions do not feel any difference from those used in conventional virtual globe or map software, meaning that the user of this algorithm can use both functions without becoming aware of any difference between them. However, an unnatural feeling might arise if the user were to zoom out after commanding a rotation because no compensation is applied to the image during zoom-out, which means that north is no longer at the top of the screen, as shown in Fig. 5. Depending on circumstances and data, this may not be a problem, but some data types are best displayed on a conventional vertical pattern of latitude and longitude lines. In such cases, during zoom-outs above a certain level, compensation is applied to bring the rotation matrix close to \mathbf{I} .

The camera location is fixed during transformation (c) in this software. Some extant map software packages display 3D images for elevation, buildings and other objects, but this method does not currently permit inclined viewing of the terrain. While this avoids complications associated with Guaranteed Visibility during user image manipulation, allowing the user to vary the camera position to provide 3D viewing of the terrain is expected to be a key feature of interactive cartographical software in the future.

4. Guaranteed Visibility

To be able to compare two geographically distant locations, one must be able to carry out measurements, observations and surveys. A widely expressed requirement emphasized in interviews with respondents was for the ability to compare any location at a closely detailed level. This algorithm can always display onscreen a location of interest to the user; therefore, it satisfies the above requirement. In this algorithm, the region of interest is indicated by dragging it with the mouse. If the screen cannot show the entire target region, for example, due to use of the zoom, the algorithm automatically divides the display to show the entire region.

4.1 Selecting Rectangular Areas

The user can designate points in the display with the mouse pointer. These locations are defined with position vectors in the client coordinate system, and the position vectors in all of the other coordinate systems are obtained using Transformations (e)-(h). These position vectors allow the user to interact with the software using the mouse pointer.

In this software, rectangles are used to select regions. A region that has been chosen by dragging the mouse to form a rectangle around it is re-drawn on the display, following globe rotations and changeover to map view, and is not displayed in the rectangle. This is one of the factors complicating the user's recognition and manipulation of regions.

This algorithm does not handle the rectangle designating the target region as a strict range specifier: the rectangle is assumed to be a guide that incorporates some level of error. The target is always handled as a rectangle in the client coordinate system. Specifically, the five points at the corners and the center of the rectangle are employed for compensation while writing the screen to preserve the rectangular shape. There is also automatic compensation for writing a rectangle in the vicinity of a pole, where the size of rectangles tends to be exaggerated.

It is technically possible to use ovals, crosses or freehand shapes to select regions, but we decided to limit the shape to rectangles because it is simpler to designate and manipulate regions with rectangles; and nested rectangles are easily perceived by the viewer.

Shared data in tree structure: M_I, α, m
 Data in nodes: M_U, P_{C0}, P_{A0}
 (M_A depends on P_{A0} and α)

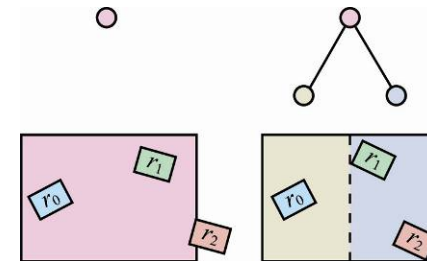


Figure 10. Display splitting directed by tree-structured data.

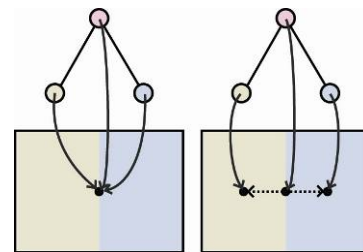


Figure 11. Seamless change of view. Child nodes shift position of P_{C0} .

4.2 Split Display

This section describes the algorithm employed for splitting the display. Reference [2] explains an approach for division of a display that cannot directly be used with the software described here. In the approach proposed in [2], the triggers for division are the line segments connecting control points. However, this presents certain issues, specifically that (1) hierarchical division cannot be expressed, (2) the method is limited in application to 2D spaces such as images, and (3) re-combination is not very stable. In order to get around the above problems, the proposed method employs a tree structure for screen division.

Let $R = \{r_0, \dots, r_{n-1}\}$ be the set of regions created by the user. Guaranteed Visibility ensures that all of the elements of R are displayed on the monitor.

First of all, it is verified whether region r_i is displayed or not using the following.

$$J = J_1 \& (J_2 | J_3) \tag{6}$$

J_1 indicates whether or not r_i is facing "toward" the viewer; J_2 indicates whether the center of r_i is within the client coordinate system; J_3 indicates whether r_i in the client coordinate system contains P_{C0} (the center of the display).

When Eq. (6) is false, it is concluded that region r_i is not being displayed, and the parameters are compensated. The system automatically inserts an emulated mouse drag operation to shift region r_i onto the viewable area.

If the automatic mouse-drag compensation fails to resolve the problem of the missing region, the display is split, either in the horizontal or the vertical direction, with a 1:1 ratio of the areas of each side.

Display splitting is expressed as tree-structured data (Fig. 10). The nodes are the set of coordinate system transformation parameters.

Splitting is done in the following steps.

- The leaf node nearest to r_i (not being displayed region) is used as the parent node.
- 2 child nodes are created.
- The child nodes are compensated to show all members of R .

The change from one display mode to another by creation of nodes is a discontinuous process, but an animation is shown during this procedure that makes it look continuous. First, to show a display identical to the parent node, the initial values for the child nodes are set as equal to the values for the parent node. Next, each child node value is smoothly varied to end with the correct value for the corresponding final child node (Fig. 11). This means that P_{C0} for each child node is initialized at the parent node value; each P_{C0} is then shifted to the center of its screen. To avoid scrolling, P_{A0} and M_U are calculated during the revaluations of P_{C0} . P_{A0} is moved at exactly the same speed as P_{C0} . M_U is rotated to place P_{C0} at the center of the Earth. Figure 12 shows an example of this animation.

If the region indicated by the child node can also be displayed by the parent node, the child node is deleted and the split screen is re-joined. In this case, P_{C0} of the child node is moved to the center of the display of the parent node in an animation that makes the process seamless.

4.3 Assisting with Context Awareness

In the previous section, we created a multi-focus display by splitting the screen to simultaneously show geographically separate locations. It has been pointed out that one of the risks of a split display is the loss of context awareness [1], but the Halo [16] could be used to provide clues about the relative positions of features and to emphasize the relationships between entities in the display.

A Halo warns of the presence of a nearby, but off-screen target, by placing a circle around it that suggests to the viewer the direction and distance to the target. Baudisch and Rosenholtz developed software for examining a 2D map in a single window and showed that it was more effective than a simple arrow [16]. Their concept was applied in this algorithm (Fig.13, Left). The center point in the display P_{C0} or region R is selected and a circle is drawn whose radius is the distance from the center of the target to the edge of the viewable area. A consistent color is used for Halos even if there are multiple targets. When two targets are on exactly opposite sides of the Earth, the circle is centered “inside” the display (Fig.13, Right).

4.4 Incorporation of Guaranteed Visibility

Since a limited degree of error is permissible in the selection of target regions with rectangles, it was decided to always use rectangular regions for display and manipulation functions in the client coordinate system. Designation of regions using rectangles was not considered to generate errors or produce other negative results. In addition, the use of rectangles facilitates the comprehension of spatial data and is intuitive in 2D operations (Fig. 12), and nesting new rectangles inside previously displayed rectangles is straightforward from the point of view of software design. Special shapes are effective as metaphors guiding to multi-dimensional data sets.

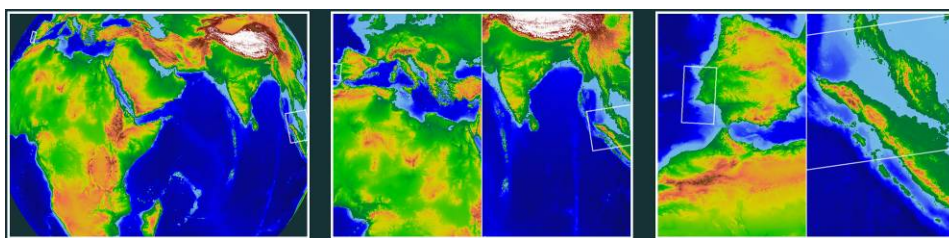


Figure 12. Example of Guaranteed Visibility in this software.

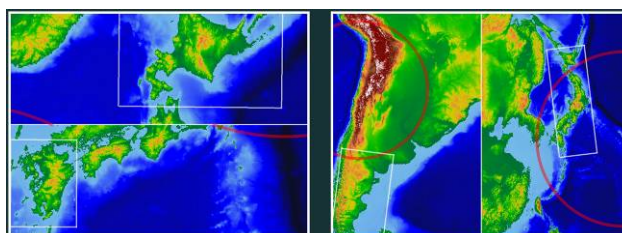


Figure 13. Example of display of circles indicating off-screen target.

This software features multi-focus views for Guaranteed Visibility, with parameter compensation for the series of coordinate systems and split displays. The user must be able to perform simultaneous measurements, observations and surveys to compare geographically separate regions. Our interviews showed a particularly strong demand for functionalities allowing detailed comparison of any user-defined region; this software is able to satisfy that demand. However, when using a split display, the viewable area in each portion of the screen is reduced, which could make it impossible to view some regions and cause the algorithm to split the display further. Since the maximum number of windows in a split display is equal to the number of areas selected, it is not practical to allow the user to select too many regions. To ensure that the viewable area does not become visually confusing, it will be necessary to try several methods to find an optimal solution, for example, assigning data to a hierarchy, selecting effective rectangles, or use of distortion.

The ratio between window sizes in the split display of MIKAN GLOBE MT has a default setting of 1:1. While this ratio can be altered, this was not found to be very effective. In this algorithm, the screens created by division are rectangular and are displayed in fixed positions; however, if it were possible to automatically optimize the shape and location of the screens, then this would be beneficial from the viewpoint of resource utilization. Future research will address this point. Another anticipated development is a tool permitting the user to make spontaneous settings for splitting windows and controlling motions.

Context-awareness using Halos for splitting the display can employ the relative locations of targets for placing the windows in multi-focus views, but the user will need to become accustomed to the algorithm to understand the relationship between locations. The centers of the display \mathbf{P}_{C0} or region R are employed as the target in the examples presented here. R operates in more straightforward fashion than \mathbf{P}_{C0} because of the difference in the extent of target movement due to user operation. In actual use, whenever any of the windows is moved, \mathbf{P}_{C0} and the Halos undergo large shifts in position, resulting in a large burden on the user who must then search the screen for the targets.

5. MIKAN GLOBE MT

MIKAN GLOBE MT is the name of the software in which the above algorithm is realized. *Mikan* is Japanese for “Mandarin orange”: it was chosen because the rind of this fruit can be peeled off in any direction. Several versions of MIKAN GLOBE are available (project URL: <http://www.mikan-globe.net/>). The code name for the version described in this study is “MT (MIKAN Technology)”. This section describes some of the specifications of MIKAN GLOBE MT.

5.1 Structure of and Searches in Surface Data

This paper employs the SRTM30 PLUS dataset, which was created based on the Digital Elevation Model (DEM) of the University of California at San Diego (UCSD) [23]. This dataset is for the entire globe and has a moderate level of accuracy. SRTM30 PLUS is a 30-seconds-of-angle mesh created from data collected by the Space Shuttle and includes an added DEM of the sea floors. A single sample covers a square approximately $900 \text{ [m]} \times 900 \text{ [m]}$ and contains 432000 [Samples] in the longitudinal direction and 216000 [Samples] in the latitudinal direction. The precision in elevation is 1 [m].

The data are divided into tile images of size 256×256 pixels and layered by generating images with half the precision. A pyramid of nine hierarchies for which the summit has one image is generated. This algorithm searches for data of interest from the pyramid to be shown in the display, as in a general Level of Detail (LOD) procedure. During this process, because of the high DOF of this system, it is sometimes difficult to check exactly how the image from inside the pyramid is transformed to the display coordinate system. In this algorithm, the hierarchical level is set at floor (λ/m) and the data on this level is shown on the display. Here, λ is the size of a single pixel on the display. The decision on whether or not to show it is performed by selecting a location in the Display Coordinate System at random and performing transformations (e)-(h) to obtain the corresponding datum. The amount of data that can be held in memory is limited, so it is essential to exclude unnecessary data whenever possible. Generally, data judged to be geometrically distant are excluded, but in this algorithm, the technique of excluding those images that had been left undisplayed for the longest periods of time turned out to be more effective than the standard of geometrical distance. This determination can be carried out by preserving the time stamps of the most recent display of images that have been read into memory.

5.2 Rendering

The rendering in the series of coordinate systems in Fig. 2 is written with data found by transformations (e)-(h) using a pixel shader. Elevation data are written using a color table.

6. Experimental Results and Evaluation

The above algorithm was implemented on a desktop PC (CPU, Intel Core2 Duo 3.0 GHz; GPU, NVIDIA GeForce 8600 GT; RAM, 2GB). The interaction was performed at the allowed speed. No effect on performance was found to occur regardless of how many divisions the display had been split into or after deliberate increases in the number of regions. Toggling between projections and the results in incorporating Guaranteed Visibility are discussed in sections 3.4 and 4.4, respectively. This section describes the user evaluations.

Users were asked to evaluate the effectiveness of MIKAN GLOBE MT. Their opinions were solicited in interviews and are summarized in this section. The software and the concepts described in this paper were explained to the test subjects before they were asked to explore with the software. Afterwards, respondents were asked for frank assessments on a scale of 1–5. Table 2 shows the questions and their corresponding scores. Test subjects consisted of 10 adults (9 men and 1 woman) in their twenties or thirties. All had experience with using basic map software.

Significance of changes in projection: Usage modes with differing values for map projection parameter α were created so that the users could experience them directly: Globe mode (with α fixed at $\sin^{-1}(0.5)$), map mode (with α fixed at 0), and switching mode (with α as described in Eq. (4)). The map mode used both equidistant cylindrical projection and azimuthal equidistant projection. The mean score for “[Q1] Overall, did you feel that it was helpful to be able to toggle between projections?” was 4.3; many respondents found this function helpful. The users also noted differences between the modes. The difference between globe mode and the cylindrical map-switching mode ([Q2]) was assigned a score of 4.3. The difference between the cylindrical map mode and cylindrical map-switching mode ([Q3]) was assigned a score of 4.6. The difference between the globe mode and azimuthal map-switching mode ([Q4]) was assigned a score of 4.3 points, and the difference between azimuthal map and azimuthal map-switching mode ([Q5]) was assigned a score of 4.2 points. These results indicate that the perception of the differences between the modes among users was high.

Mouse operations and guaranteed visibility: Subjects were asked to create multiple regions by dragging the mouse and to

pan and zoom to verify that the regions continued to be visible. They were then asked to enter the cylindrical map-switching mode, which assigns Pan and Rotate to the same mouse button (Section 3.2) and transforms Pan operations to Rotate operations (Section 3.3). Users were then asked to compare the ease of use of this feature with conventional mapping and geographic software packages ([Q6]). The mean score was 4.0, with most of the subjects considering the operation to be natural, but some feeling that the operation would take some getting used to. The ease of region designation ([Q7]) was assigned a score of 4.5, and most of the subjects agreed that the error associated with selecting rectangular areas was not particularly problematic. In response to the question of whether the software operated in accordance with their intentions when the Guaranteed Visibility function was active ([Q8]), the mean score was 4.4, indicating that this algorithm was effective.

Halo function: The Halo function was activated and the processes described in the preceding test were repeated. On this occasion, the functions were assessed differently by the subjects. When the target was the center of the display, P_{C0} ([Q9]), the score was 2.6, but when the target was the region R ([Q10]), the score was 3.7. If R is used, there is no need to use P_{C0} . Some subjects also noted that the zoom function provided an overview of the complete region, which meant that neither P_{C0} nor R was considered necessary.

The reason Halos were not considered to be very effective appears to have been because it is difficult for the user to formulate a sense of where the 2D information Halo data is displayed in a high-DOF map. The users’ opinions also indicated that the placement of locations marked by Halos were not very effective in the split displays. It may therefore be better to use absolute expressions, for example, to create an overview object

Table 2. Questions for user evaluations

Questions	Score	Comments
[Q1] Overall, did you feel that it was helpful to be able to toggle between projections?	4.3	1: It was unhelpful. 3: Not sure. 5: It was helpful.
[Q2] Do you feel that it was helpful to have the cylindrical map-switching mode in comparison with globe mode?	4.3	
[Q3] Do you feel that it was helpful to have the cylindrical map-switching mode compared to having the cylindrical map mode?	4.6	
[Q4] Do you feel that it was helpful to have the azimuthal map-switching mode compared to having the globe mode?	4.3	
[Q5] Do you feel that it was helpful to have the azimuthal map-switching mode compared to having the azimuthal map mode?	4.2	
[Q6] Did you feel that MIKAN GLOBE MT was easier to use than previous mapping and geographic software?	4.0	1: Strongly disagree. 3: Not sure. 5: Strongly agree.
[Q7] Was it easy for you to designate regions of interest?	4.5	
[Q8] Were you able to perform operations as intended while Guaranteed Visibility was activated?	4.4	1: It was unhelpful. 3: Not sure. 5: It was helpful.
[Q9] Do you feel that it was helpful to have the Halo function when the target was the center of the display, P_{C0} ?	2.6	
[Q10] Do you feel that it was helpful to have the Halo function when the target was region R ?	3.7	1: Strongly disagree 3: Not sure. 5: Strongly agree.
[Q11] Overall, do you think MIKAN GLOBE MT has potential for use as a new mapping tool?	4.4	
[Q12] Would you use MIKAN GLOBE MT?	4.2	

from a world map showing the currently displayed location.

Overall impressions: The question “Overall, do you think MIKAN GLOBE MT has potential for use as a new mapping tool? ([Q11])” was scored 4.4, and “Would you use MIKAN GLOBE MT? ([Q12])” was scored 4.2. The subjects liked the ability to create maps that are not available on current software or paper media, and the ability to manipulate distant locations, as well as other features.

7. Conclusion

This paper described the design of map software that satisfied several currently unfilled needs that were elicited through a questionnaire survey of map software users. This design employed a new algorithm which differs from the direct browse style of geometric space in currently available packages.

The most notable feature of this algorithm is that it can handle geographical data in both 2D and 3D. It also offers the user with a high-DOF environment that facilitates the use of a wide selection of features that can be displayed in accordance with the needs of the user. This study does not describe research on optimal projection methods or parameter selection as future research will address issues related to finding optimal solutions to specific cases in a high-DOF environment.

This paper enabled the display of global views and the deformations that are encountered with different map views. Future investigations by the authors will deal with similar strategies to optimize the visualization of object information.

References

- [1] Elmqvist, N., Henry, N., Riche, Y. and Fekete, J.-D. 2008. Melange: Space Folding for Multi-Focus Interaction. In Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems, 1333-1342.
- [2] Shoemaker, G. and Gutwin, C. 2007. Supporting Multi-Point Interaction in Visual Workspaces. In Proceedings of the ACM CHI 2007 Conference on Human Factors in Computing Systems, 999-1008.
- [3] Goetz, F., Boening, B., Domik, G. 2004. Mapping High Quality Textures on an Interactive Globe. In Proceedings of the Fourth IASTED International Conference VISUALIZATION, IMAGING, AND IMAGE PROCESSING. 619-624.
- [4] Ueda, M. 2007. Geographical Relations Dynamics (1) on an Interactive 3D CG Digital Globe. In Proceedings of the 3rd IASME/WSEAS Int. Conf. on Energy, Environment, Ecosystems and Sustainable Development. 243-248.
- [5] Companje, R., Dijk, N., Hogenbirk, H., Mast, D. 2006. Globe4D, time-traveling with an interactive four-dimensional globe. In Proceedings of the 14th annual ACM international conference on Multimedia. 959-960.
- [6] Benko, H., Wilson, A. D., Balakrishnan, R. 2008. Sphere: Multi-Touch Interactions on a Spherical Display, In Proc. UIST 2008. 77-86.
- [7] Munzner, T., Guimbretiere, F., Tasiran, S., Zhang, L., and Zhou, Y. 2003. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. In Proceedings of ACM SIGGRAPH 2003, 453-462.
- [8] Furnas, G.W. and Bederson, B.B. 1995. Space-Scale Diagrams: Understanding Multiscale Interfaces. In Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems, 234-241.
- [9] Wijk, J.J. and Nuij, W.A.A. 2003. Smooth and Efficient Zooming and Panning. In Proceedings of the IEEE Symposium on Information Visualization, 15-22.
- [10] Igarashi, T. and Hinckley, K. 2000. Speed-Dependent Automatic Zooming for Browsing Large Documents. In Proceedings of the ACM Symposium on User Interface Software and Technology, 139-148.
- [11] Appert, C. and Fekete, J.-D. 2006. OrthoZoomscroller: 1D Multi-Scale Navigation. In Proceedings of the ACM CHI 2006 Conference on Human Factors in Computing Systems, 21-30.
- [12] Furnas, G.W. 1986. Generalized Fisheye Views. In Proceedings of the ACM CHI'86 Conference on Human Factors in Computer Systems, 116-123.
- [13] Furnas, G.W. 2006. A Fisheye Follow-up: Further Reflections on Focus+Context. In Proceedings of the ACM CHI 2006 Conference on Human Factors in Computing Systems, 999-1008.
- [14] Sarkar, M., Snibbe, S.S., Tversky, O.J., and Reiss, S.P. 1993. Stretching the Rubber Sheet: A Metaphor for Visualizing Large Layouts on Small Screens. In Proceedings of the ACM Symposium on User Interface Software and Technology, 81-91.
- [15] Perlin, K. and Fox, D. 1993. Pad: An Alternative Approach to the Computer Interface. In Proceedings of Computer Graphics (SIGGRAPH93), 57-64.
- [16] Baudisch, P. and Rosenholtz, R. 2003. Halo: A Technique for Visualizing Off-Screen Objects. In Proceedings of the ACM CHI 2003 Conference on Human Factors in Computing Systems, 481-488.
- [17] Zellweger, P.T., Mackinlay, J.D., Good, L., Stefik, M., and Baudisch, P. 2003. City Lights: Contextual Views in Minimal Space. In Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems, 838-839.
- [18] Gustafson, S.G. and Irani, P.P. 2007. Comparing Visualizations for Tracking Off-Screen Moving Targets. In Extended Abstracts of the ACM CHI 2007 Conference on Human Factors in Computing Systems, 2399-2404.
- [19] Irani, P., Gutwin, C., and Yang, X.D. 2006. Improving Selection of Off-Screen Targets with Hopping. In Proceedings of the ACM CHI 2006 Conference on Human Factors in Computing Systems, 299-308.

[20] Kopf, J., Uyttendaele, M., Deussen, O., and Cohen, M. 2007. Capturing and Viewing Gigapixel Images. ACM Transactions on Graphics, 26, 3. Article 93. 1-10.

[21] Matsuyama, K., Okamoto, M., MIKAN GLOBE MT: An Interactive World Map featuring Seamless Map Projection. Nicograph 2008.

[22] Snyder, J.P. 1987. Map Projections: A Working Manual, Vol. 1395, U.S. Geological Survey Professional Paper, U.S. Government Printing Office Washington D.C..

[23] SRTM30_PLUS.

http://topex.ucsd.edu/WWW_html/srtm30_plus.html.