The Journal of the Society for Art and Science Vol. 14, No. 3, pp. 46 – 56

MapSlider: A Property Based Interface for World Map Software

Katsutsugu Matsuyama(member) **Kouichi Konno**(member)

Iwate University

kmatsu (at) iwate-u.ac.jp

Abstract

By focusing on the properties used to measure map projections and utilizing them to manipulate map projection parameters, we have developed a new interface for manipulating such map projections. With our interface, a user can create his or her own map projections by manipulating the weight of each property via sliders. Specifically, we employ equidistancy, equiareality, and conformality as property elements and allow the user to balance these property elements according to their perceived levels of importance. Additionally, along with this interface, we also have developed a technique for reducing the number of projection parameters by assuming that latitude lines are horizontal and that longitude widths are identical in each latitude, which improves execution speed. Furthermore, as an additional interface, we have implemented a method for creating interrupted maps. This allows users to make interruptions on a map easily by clicking the interruption panel. Our system also provides a new manipulation method with which the user can traverse a number of existing cylindrical projections, such as those created with the Mercator and Lambert cylindrical projection methods.

1 Introduction

Interactive world map applications are among of the most commonly used types of software. Most of these software packages in current use show geographical features mapped onto the surface of a sphere or flat surface. In such cases, a projection method is needed to express geographical features on a flat surface.

All map projections distort the surface in some fashion. Different map projection methods exist in order to preserve some properties of a sphere-like body, while permitting an acceptable level of distortion in other properties, depending on the purpose of the map. To accomplish this, cartographers have developed various projection methods that trade off different distortion types [21].

We can evaluate map projection methods from the viewpoint of what is necessary to preserve:

- Distance from specific points or a line (equidistant)
- Area (equiareal)
- Local shape (conformal)

These categories are most often used to classify the properties of map projection methods.

Many of the current map software applications use the single map projection method. Although three-dimensional (3D) globe software expresses the Earth realistically, it cannot display all of the Earth ' s surface at the same time, which means a user needs to rotate the globe in order to see the opposite side. Furthermore, such projections usually only provide a single viewpoint, and are unable to display multi-view images.

Previous research efforts have resulted in the development of mapping systems that allow the user to manipulate map projection parameters. These include compositing existing map projections according to the context of a viewpoint [10], allowing the user to specify projection parameters numerically [8, 9, 11], and transiting seamlessly between a world map and a globe [14].

In this study, we developed a new interface that allows map projections to be manipulated interactively. We accomplished this by focusing on the properties used to measure map projections and using them to create an interface for manipulating map projection parameters. With our interface, a user can create his or her own map projections by manipulating the weight of each property via sliders. Specifically, we employ equidistancy, equiareality, and conformality as the property elements. This interface allows users to balance the property elements according to their perceived level of importance and differs from existing methods that only allows users to specify geometric projection parameters numerically.

Since our developed system performs projection parameter calculations by solving an optimization problem, the number of parameters requiring manipulation affects the execution speed. Accordingly, in an effort to improve execution speed, we also developed a technique for reducing the number of projection parameters by assuming that latitude lines are horizontal and that longitude widths are identical in each latitude.

Furthermore, as an additional interface, we have implemented a method for creating interrupted maps. Interrupted maps, such as the Goode homolosine projection, are useful when preserving partial directions. With our interface, users can easily make interruptions on the map by clicking the interruption panel.

By using a property based interface, map projection selections can be considered on a functional basis. This provides users with a different experience from conventional interfaces that specify properties numerically. Our techniques can be applied widely to a variety of world map software applications, and can even be used educationally to promote understanding of basic map projection methods. Furthermore, it may also be possible to use our techniques as input to other methods that create composite map projections.

This paper is made corrections reflecting referee's comments of the manuscript submitted to NICOGRAPH International 2014 [13].

2 Related Work

When transforming the latitudes and longitudes of locations on the surface of a sphere or ellipsoid into locations on a plane, a map projection method is required. To provide this, cartographers have developed various projection methods that trade off different distortion types [21].

Some research efforts have resulted in mapping systems that allow the user to manipulate map projection parameters. For example, there are systems that allow the user to specify projection parameters numerically in order to create flexible projections [8, 9, 11]. However, because the above mentioned systems specify geometric parameters numerically, and because the connections between parameters and properties are weak, users are usually unable

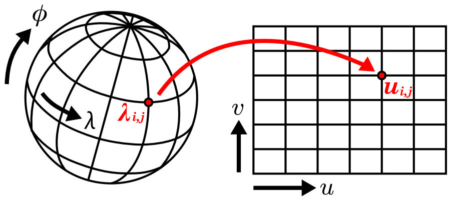


Figure 1: A mapping from the sphere $\boldsymbol{\lambda} = (\lambda, \phi)$ into a planar domain $\boldsymbol{u} = (u, v)$.

to easily comprehend how parameter changes will affect map properties.

Jenny developed a software application that composites existing map projections according to the context of a viewpoint [10], and Matsuyama and Okamoto developed MIKAN GLOBE, a software application that allows the user to traverse seamlessly between existing map projections and a globe [14]. However, while these applications allow users to change the existing map projection continuously according to a specific viewpoint, they do not provide tools that can be used to create original map projections.

Map interaction applications make continuous visual changes to different maps using techniques such as morphing [16], realizing Focus+context using folding [4], a fish-eye lens [5, 6], and the rubber sheet distortion [17]. There are also techniques that enable interactions with a globe using a spherical display [3, 1].

Additionally, there are methods that unfold 3D geometry by considering conformality in order to automatically generate texture atlases [12] and to display wide-angle images [2]. Deformation techniques are often used to maintain local conformality [7, 18]. However, since these studies do not take equidistancy and equiareality into consideration, they cannot be directly employed for our purpose.

3 Map Projection Properties

Map projections transform the latitudes and longitudes of locations on the surface of a sphere or ellipsoid into locations on a plane. When drawing the Earth on a plane, distortions occur and distances, areas, and angles cannot be displayed correctly simultaneously. Different map projection methods have been devised to preserve certain properties while allowing acceptable levels of distortion in others, depending on the purpose of the map, and cartographers have developed various projection methods that trade off different distortion types [21]. These various map projection techniques are classified and evaluated in terms of equidistancy, equiareality, and conformality. In this section, we will briefly explain these properties (see [19, 20, 21] for more detail).

In this research, we consider the Earth as a sphere instead of an ellipsoid (as is common in most geographic coordinate systems). Additionally, we define a projection as a mapping from the sphere parametrized by longitude λ and latitude ϕ into a planar domain parametrized by u and v (Fig. 1). We also define parameters $\boldsymbol{\lambda} = (\lambda, \phi)$ and $\boldsymbol{u} = (u, v)$ in vector form, in which a mapping can be represented as $u(\lambda, \phi)$ and $v(\lambda, \phi)$, or $\boldsymbol{u}(\boldsymbol{\lambda})$. For simplicity of explanation, we will assume the radius of the Earth as $R = 1$ and drop R for the remainder of this paper.

Equidistant

A projection is equidistant when it preserves distance from a specific point or from a line. Generally, the projective center becomes the specific point in an azimuthal projection and the equator becomes the specific line in a cylindrical projection. Azimuthal equidistant projection and equidistant cylindrical projection are representative of the equidistant projection method.

The projection formula of an equidistant cylindrical projection is

$$\begin{aligned} u(\lambda, \phi) &= \lambda \\ v(\lambda, \phi) &= \phi \end{aligned} \tag{1}$$

In this paper, we assign the equator (great circle of latitude 0) to be the specific line. This then becomes similar to a cylindrical projection to which the equator is in constant contact with the circle.

Equiareal

A projection is equiareal if the projection preserves area. Concretely, a small area on a sphere will be represented by an equal sized area on a map. Lambert cylindrical equal-area projections and Lambert azimuthal equal-area projections are representative of the equiareal projection method.

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

Figure 2: The user interface consists of (a) map viewer, (b) slider, (c) interrupter, and (d) distance viewer. (a) displays the map projection result. The user can change the map projection through (b) by changing the weight value of each property. (c) allows the user to make an interrupted map by clicking on it. (d) is a tool for monitoring the distortion of each property.

Conformal

A projection is conformal if the projection preserves shapes locally.

We can describe the local properties of such a mapping in terms of differential north and east vectors of the projection $\boldsymbol{h}$ and $\boldsymbol{k}$ [20, 2].

$$\boldsymbol{h} = \begin{bmatrix} \frac{\partial u}{\partial \phi} \\ \frac{\partial v}{\partial \phi} \end{bmatrix}, \boldsymbol{k} = \begin{bmatrix} \frac{\partial u}{\partial \lambda} \\ \frac{\partial v}{\partial \lambda} \end{bmatrix} \frac{1}{\cos(\phi)} \tag{2}$$

A mapping is conformal if $\boldsymbol{h}$ is a rotation of $\boldsymbol{k}$, such that

$$\boldsymbol{h} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \boldsymbol{k} \tag{3}$$

or, equivalently

$$\frac{\partial u}{\partial \phi} = -\frac{\partial v}{\partial \lambda}\frac{1}{\cos(\phi)}, \frac{\partial v}{\partial \phi} = \frac{\partial u}{\partial \lambda}\frac{1}{\cos(\phi)} \tag{4}$$

These are the Cauchy-Riemann equations used for mapping a sphere to a plane [20, 2]. The angle between two small vectors from the arbitrary points on Earth is expressed at the same angle on a map.

The Mercator projection is a conformal cylindrical projection. If $u = \lambda$ then $\frac{\partial u}{\partial \lambda} = 1$, which implies that

$$\frac{\partial v}{\partial \phi} = \frac{1}{cos(\phi)}, v = \ln(\sec\phi + \tan\phi) \tag{5}$$

Google Maps and Microsoft's Bing Maps use the Mercator projection method. However, Mercator projections have enormous areal distortion at higher latitudes, which makes it impossible to show the entire globe because the poles are placed at an infinite distance from the equator.

4 Our system

In this section, we provide details of our map projection manipulation system. First, we will provide an overview of our newly developed interface (section 4.1) and the technique used for realizing the interface (section 4.2). We calculate projection parameters by solving optimization problem. To improve execution speed, a technique for reducing the number of projection parameters is introduced (section 4.4). We then describe an additional interface used for creating interrupted maps, along with a method for displaying such maps (section 4.5).

Mapping is discretized by sampling a uniform grid in (λ, ϕ) indexed by integers (i, j) (Fig. 1). We define V as the entire set of vertices (i, j), $0 \leq i \leq N$ and $0 \leq j \leq M$. For each spherical grid vertex $\boldsymbol{\lambda}_{i,j}$, we compute the value of its corresponding $\boldsymbol{u}_{i,j}$ in two dimensions via an optimization operation.

4.1 User Interface

Fig. 2 shows our user interface. The top-level layout of the user interface consists of a map viewer, slider, interrupter, and distance viewer. The map projection result is displayed on the map viewer. The slider can be used to change the weight values of equidistancy, equiareality and conformality continuously from 0 to 1. This interface allows the user to balance the property elements according to their perceived level of importance. An interrupter is an interface used for making an interrupted map. This interface allows the user to cut along the vertical line from the selected point. Those cut lines are then expressed as lines on the interface. The user can change on/off of cutting by clicking. The distortion of each property is displayed on the distance viewer.

4.2 Property Energies

Equidistancy

We evaluate the equidistancy of a projection using the equidistant property described in section 3. Our system measures how far a projection is from the equidistant state. Since we assume that equidistancy evaluates the distance from the equator, we define an energy (penalty) via the following equation:

$$E_d = \sum_{(i,j)\in V} (v_{i,j} - \phi_{i,j})^2 \tag{6}$$

This equation implies that equidistancy only evaluates the vertical component.

Equiareality

As with equidistancy, we define equiareality as the difference between a projection and the equiareal state. The area of an arbitrary discretized patch on a sphere is

$$S_{i,j} = (\phi_{i,j+1} - \phi_{i,j})(\lambda_{i+1,j} - \lambda_{i,j}) \cos\phi_{i,j} \tag{7}$$

We define an energy as the area difference between a patch and the quadrilateral to which it is projected.

$$E_a = \sum_{(i,j)\in V} (S_{i,j} - A_{i,j})^2 \tag{8}$$

where $A_{i,j}$ is the area of a quadrilateral $\boldsymbol{u}_{i,j}$, $\boldsymbol{u}_{i+1,j}$, $\boldsymbol{u}_{i,j+1}$, and $\boldsymbol{u}_{i+1,j+1}$.

Conformality

We define conformality (Fig. 3) by discretizing the equations (4) [2], giving

$$\begin{aligned} u_{i,j+1} - u_{i,j} &= -(v_{i+1,j} - v_{i,j})/\cos\phi_{i,j} \\ v_{i,j+1} - v_{i,j} &= (u_{i+1,j} - u_{i,j})/\cos\phi_{i,j} \end{aligned} \tag{9}$$

To consider the weight balance with equiareality (equation (8)), the area of each patch is multiplied to each side of equation (9). Since the area of a patch on a sphere is proportional to $\cos(\phi)$, we multiply each side by $\cos(\phi)$. As a result, we define an energy using the following equation:

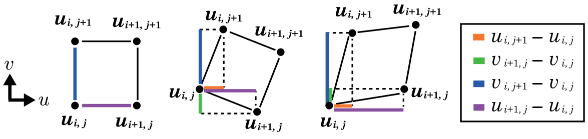


Figure 3: Conformality. (a) and (b) are conformal and eq. (9) is satisfied. (c) is not conformal and conformality (eq. (10)) is more than 0.

$$\begin{aligned} E_c = &\sum_{(i,j)\in V} ((v_{i+1,j} - v_{i,j}) + \cos\phi_{i,j}(u_{i,j+1} - u_{i,j}))^2 \\ + &\sum_{(i,j)\in V} ((u_{i+1,j} - u_{i,j}) - \cos\phi_{i,j}(v_{i,j+1} - v_{i,j}))^2 \end{aligned} \tag{10}$$

4.3 Fixation Restrictions and Exception-handling

This section describes fixation restrictions and exception-handling.

To fix the center of a map to the center of a display domain, we add $\boldsymbol{u}_{N/2,M/2} = (0, 0)$ to the restrictions. Each point on the equator is also fixed so that they are located at equal intervals horizontally: $u_{i,M/2} = \pi(2i/N - 1)$. The resulting energy of the fixation restrictions is

$$E_f = |\boldsymbol{u}_{N/2,M/2}|^2 + \sum_{i=0}^{N} (u_{i,M/2} - \pi(2i/N - 1))^2 \tag{11}$$

It is also necessary to handle the exceptional cases that occur when the user sets the weight of equidistancy, equiareality, and/or conformality to 0.

If the weights of conformality and equidistancy are 0, the system has no key for handling form restrictions. In this case, we fix the horizontal component of the vertices as follows:

$$E_e = \sum_{(i,j)\in V} (u_{i,j} - \pi(2i/N - 1))^2 \tag{12}$$

This is equivalent to a Lambert cylindrical equal-area projection.

If the weights of conformality and equiareality are 0, there are no horizontal position restrictions. In this case, we fix the horizontal component of the vertices as shown in equation (12).

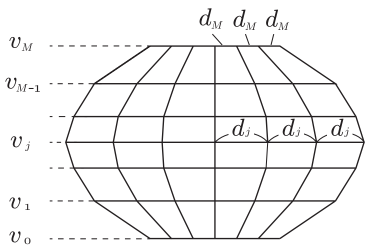


Figure 4: Reducing the number of projection parameters.

Finally, if all of the weights are set to 0, we arbitrarily display an equidistant cylindrical projection in order to prevent a visual breakdown.

4.4 Reducing the number of projection parameters

In this section we describe a technique for reducing the number of projection parameters. As we have seen with the energies discussed in the previous sections, the number of projection parameters is the order of the number of divisions, i.e., $O(NM)$. Since our newly developed system performs projection parameter calculations by solving an optimization problem, the number of parameters that must be considered has a direct affect on execution speed. Accordingly, to improve execution speed, our interface incorporates a technique for reducing the number of projection parameters by assuming that latitude lines are horizontal and that the longitude widths are identical in each latitude.

By assuming latitude lines are horizontal, $v_{i,j}$ can be considered as a parameter v_j parametrized only by j (Fig. 4). Furthermore, by assuming the width increments of longitude in each latitude are identical, we can introduce the parameter d_j to express the width increments, i.e., $d_j = u_{i+1,j} - u_{i,j}$. As a result, the number of parameters can be reduced to the order of $O(2M)$.

We then recalculate the energy formulae so that they become formulae of parameters v_j and d_j.

Equidistancy

Since $\phi_{i,j}$ vary only by j, $\phi_{i,j}$ can be considered as ϕ_j parameterized only by j, i.e. $\phi_{i,j} = \phi_j$.

Using v_j and ϕ_j, The equation (6) can be transformed as follows:

$$E_d = N \sum_{j=0}^{M} (v_j - \phi_j)^2 \tag{13}$$

Equiareality

First, since ϕ and λ are divided equally, $(\phi_{i,j+1} - \phi_{i,j}) = \pi/M$ and $(\lambda_{i+1,j} - \lambda_{i,j}) = 2\pi/N$, we can calculate the area of a part of a sphere sliced horizontally S_j from equation (7).

$$S_j = NS_{i,j} = \cos(\phi_j)\frac{2\pi^2}{M} \tag{14}$$

Next, since latitude lines are horizontal, each quadrilateral $\boldsymbol{u}_{i,j}$, $\boldsymbol{u}_{i+1,j}$, $\boldsymbol{u}_{i,j+1}$, and $\boldsymbol{u}_{i+1,j+1}$ becomes a trapezoid. The area of this teapezoid $A_{i,j}$ and sliced part A_j can be easily calculated by

$$A_j = NA_{i,j} = N(d_j + d_{j+1})(v_{j+1} - v_j)/2 \tag{15}$$

Now, the energy of equation (8) becomes

$$E_a = N \sum_{j=0}^{M} (S_j/N - A_j/N)^2 \tag{16}$$

Conformality

Since latitude lines are horizontal, $(v_{i+1,j} - v_{i,j}) = 0$. And, from the definition of d_j, we can replace $(u_{i,j+1} - u_{i,j})$ with $(i - N/2)(d_{j+1} - d_j)$. Using these replacements, we can transform equation (10) to

$$E_c = \sum_{(i,j)\in V} ((i - N/2)\cos\phi_j(d_{j+1} - d_j))^2 + \sum_{j=0}^{M} (d_j - \cos\phi_j(v_{j+1} - v_j))^2 \tag{17}$$

$(i - N/2)$ can be calculated in advance, so the first term of equation (17) can be substantially considered the form of $N\sum_{j=0}^{M}$.

4.5 Interruption

Interrupted maps are the maps cut on arbitrarily chosen lines. Cut lines are often designed to fall on less important areas e.g. oceans. In order to restrict visual distortion, we offer an interface which enables

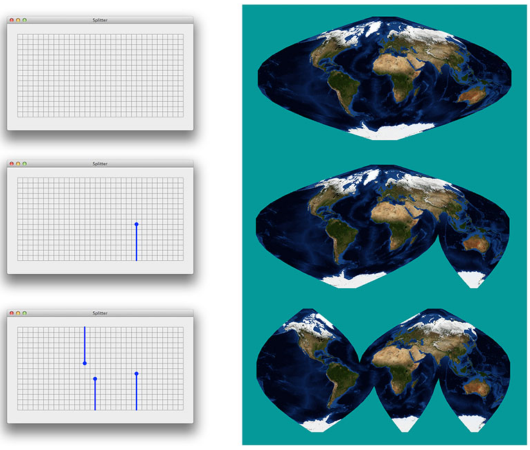


Figure 5: Our interrupter and interrupted maps.

users to create interrupted maps. When the user makes cut lines, our system renders a map which is cut by the lines. Since many existing interrupted maps have vertical cut lines, we have designed our interface which is specialized in vertical cut lines.

Our interruption panel enables users to click and change on/off of cutting (Fig. 5). When the user moves a mouse on the panel, our system draws a vertical line segment using mouse position endpoints and the peripheral of the domain. If the mouse position is in the upper half of the domain, the upper peripheral is selected. If it is in the lower half, the lower peripheral is selected (Fig. 5). Users can make multiple lines, after which the system cuts out and generates an interrupted map based on those lines.

Clicked points are stored as a point list $P = [p_0, p_1, ..., p_{Q-1}]$. From P, we make SplitTrees, tree structures to which each node expresses a range $W[i_{min}, i_{max}, j_{min}, j_{max}]$, i.e. from $\boldsymbol{u}_{i_{min}, j_{min}}$ to $\boldsymbol{u}_{i_{max}, j_{max}}$ (Fig. 6). The root node of this tree includes the equator, and two trees, north and south side, are created (j_{min} of the root node of the north side tree is $M/2$).

Using these SplitTrees, we render the projection parameters v_j and d_j. Starting from the root of the tree, our system then renders the parameters within the range W of the node.

Here, we will use the north side as an example. First, we prepare a variable u_c that expresses the center position of the u direction. The initial value is $u_c = 0$.

From each range W and the parameters v_j and d_j, we determine (u, v) as follows:

$$
\begin{aligned}
u_{(i,j) \in W} &= u_c - C_w d_j + d_j (i - W[i_{min}]) \\
v_{(i,j) \in W} &= v_j \qquad (18) \\
C_w &= (W[i_{max}] - W[i_{min}])/2
\end{aligned}
$$

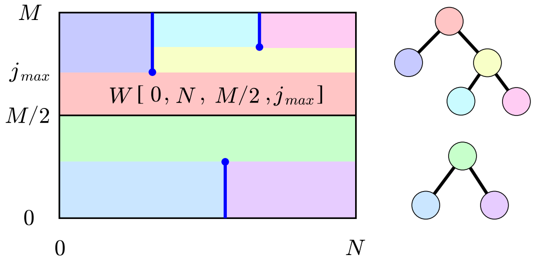


Figure 6: Rendering an interrupted map.

$W[i_{min}]$ ($W[i_{max}]$) means i_{min} (i_{max}) of W and C_w is a temporal variable which keeps the mean position for the range W.

If the node has children (equivalent to the case where $W[j_{max}]$ is not N), we calculate a new u_c (described as u_{cn}) from the range of the current node W_a and the range of the child node W_b as follows:

$$
\begin{aligned}
u_{cn} &= (u_L + u_R)/2 \qquad (19) \\
u_L &= u_c - C_w d_k + d_k (W_b[i_{min}] - W_a[i_{min}]) \\
u_R &= u_c - C_w d_k + d_k (W_b[i_{max}] - W_a[i_{min}]) \\
C_w &= (W_a[i_{max}] - W_a[i_{min}])/2 \\
k &= W_a[j_{max}]
\end{aligned}
$$

Using these W_b and new u_{cn}, a rendering is performed recursively.

4.6 Optimization

Final energy is as follows:

$$
E = w_d a_d E_d + w_a a_a E_a + w_c a_c E_c + a_f E_f + a_e E_e \qquad (20)
$$

where w_d, w_a and w_c are values which the user inputs with the slider. a_d, a_a, a_c, a_f and a_e are weight values of each term. In this paper we use $a_d = 1$, $a_a = 32$, $a_c = 32$, $a_f = 1000$ and $a_e = 1000$ (when $N = 32$ and $M = 16$). We set higher values to a_f and a_e so that these constraints act as hard constraints. We use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [15] to minimize E.

5 Results

We implemented the algorithms shown above using the Python programming language and the

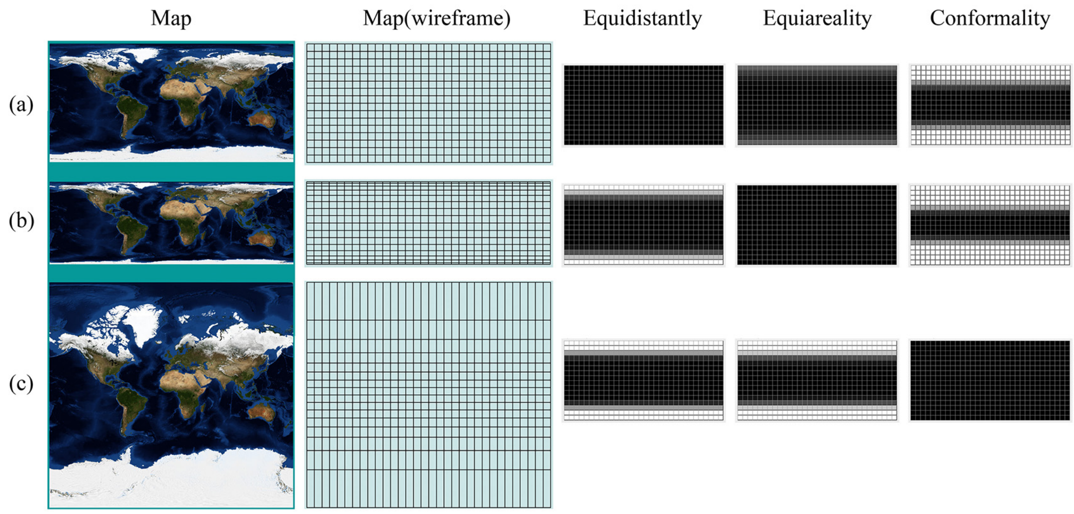


Figure 7: Projection results. Used parameters are (a) $w_d = 1, w_a = 0, w_c = 0$, (b) $w_d = 0, w_a = 1, w_c = 0$ and (c) $w_d = 0, w_a = 0, w_c = 1$.

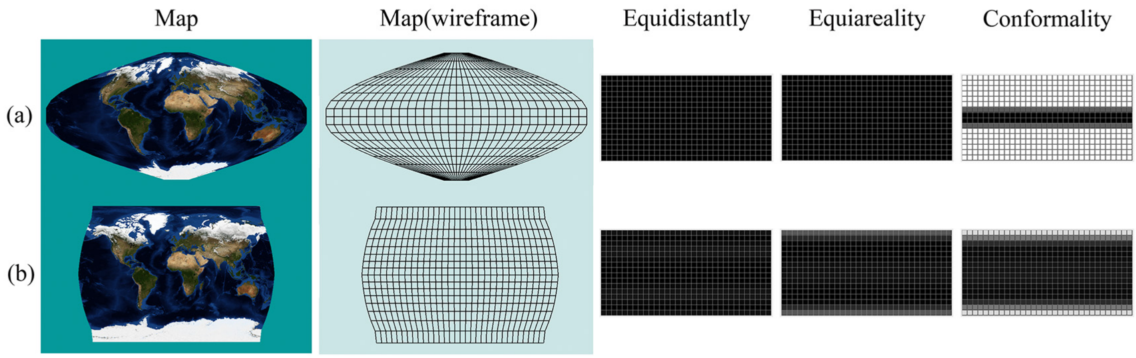


Figure 8: Projection results by the combination of parameters. Each parameters are (a) $w_d = 1, w_a = 1, w_c = 0$ and (b) $w_d = 0.86, w_a = 0.59, w_c = 0.37$.

SciPy library for optimization. Although we did not perform program acceleration, we found that the program operates interactively. We use earth textures from NASA Visible Earth (http://visibleearth.nasa.gov). Our map viewer allows the user to scale and scroll the generated map.

Fig. 7(a) shows a projection result when only equidistantly is set to 1 ($w_d = 1, w_a = 0, w_c = 0$). This projection is equivalent to the equidistant cylindrical projection. Figs. 7 (b) and (c), respectively, show the result when only equiareality and conformality are set to 1. These are, respectively, equivalent to the Lambert cylindrical projection and Mercator projection methods. Our distance viewer shows the distortion of each property. The horizontal direction indicates longitude and the vertical direction indicates latitude. The black area expresses the part without distortion, and it becomes white as distortion becomes large.

Fig. 8 shows projection results obtained by the combination of parameters. Here, we can see that shape appearance of the map changes according to the selected parameter. We also can see the degree of distortion through our distance viewer. The projection of Fig. 8(a) constricts the distortions of equidistantly and equiareality but sacrifices conformality and this is equivalent to the Sanson-Flamsteed projection. The projection of Fig. 8(b) equally constricts all the distortion.

Since particular parameter sets correspond to a

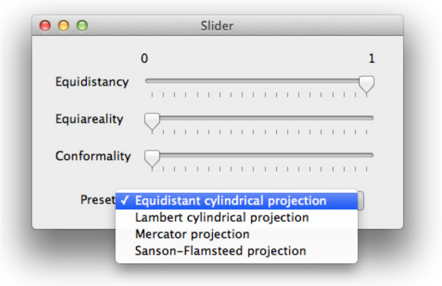


Figure 9: An extended interface.

projection method, we can add a function of choosing a projection method for setting parameters to our system. In this research, we also implemented an extended interface as shown in Fig. 9. After setting parameters through choosing preset projection method, the user can change each parameter with slides.

Fig. 5 shows interrupted maps. As discussed previously, users can easily make interruptions on the map by clicking the interruption panel. Figs. 10 and 11 are examples of different texture.

6 Discussion

Performance

The execution speed of our method is dependent on the division numbers N and M. When $N = 32$ and $M = 16$, the system operates rapidly (our videos use these values).

We executed on Mac OS X (CPU: 3.4GHz Intel Core i7, Memory: 8GB 1333MHz DDR3, GPU: AMD Radeon HD 6970M 1024MB). Table 1 shows the calculation time of our algorithm with or without reducing the number of projection parameters (Sec. 4.4). We measured convergence time from the initial state of $(w_d = 1, w_a = 0, w_c = 0)$ to $(w_d = 1, w_a = 1, w_c = 0)$. By reducing the number of parameters, we can reduce the calculation time dramatically, and can realize interactive operation. While Intel Core i7 is multi-core CPU, we did not perform program parallelization. Parallelization and the hardware utilized will also contribute to speed improvements.

Table 1: Calculation time [s]

(N, M)	with reducing	without reducing
(8, 4)	0.05	14.26
(16, 8)	0.12	7386.06
(32, 16)	0.34	24368.02

Comprehension

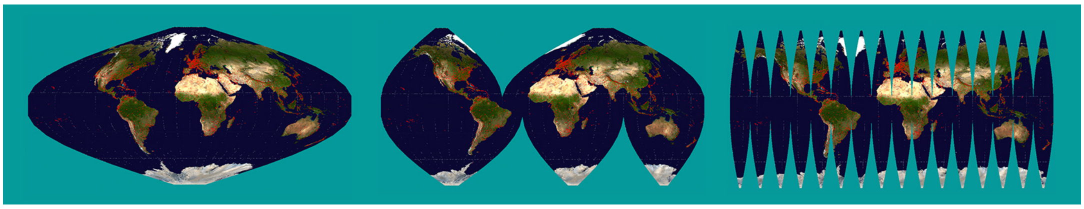
While our interface can be used to traverse some existing cylindrical projections, such as those using the Mercator projection and the Lambert cylindrical projection methods, it cannot express all cylindrical projection types. However, since the purpose of our study was simply to develop a new map manipulation method, we did not attempt to examine all projection methods. As for questions related to handling the comprehension and interface, it is expected that a more advanced system can be developed by applying our interface to other situations where the projection surface shape and/or the projection light source position are fixed.

7 Conclusion

In this study, we have developed a new interface that can be used to manipulate map projections. To accomplish this, we focused on the properties used when measuring map projections and used those properties to create an interface capable of manipulating map projection parameters. Along with this interface, we also have developed a technique of reducing the number of projection parameters by assuming latitude lines are horizontal and that longitude widths are identical in each latitude, which allows us to improve execution speed. Furthermore, we have implemented an additional interface that can be used to create interrupted maps.

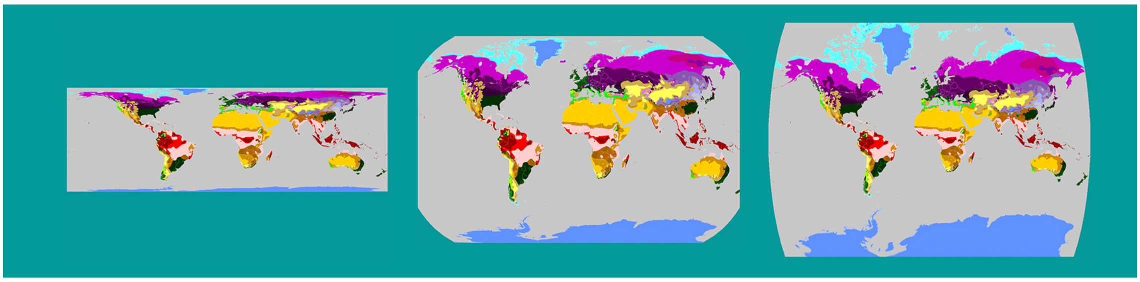
Our proposed interface can be used to balance the property elements according to their perceived levels of importance, and to make interruptions on a map by clicking the interruption panel. As a result, our system provides users with a new manipulation method that can traverse some of the existing cylindrical projections, such as those created using the Mercator projection and Lambert cylindrical projection methods.

While this newly developed interface is based on a cylindrical projection method, we are planning to



(a) (b) (c)

Figure 10: World airport map by Jpatokal. This image is licensed under the Creative Commons Attribution-Share Alike 3.0.



(a) (b) (c)

Figure 11: Global warming map by Dave Pape. This image is licensed under the Creative Commons Attribution-Share Alike 3.0.

expand it to include azimuthal- and conic-based projection methods in the future. We will also seek a way to unify our method with a 3D globe, which will allow users to see overviews and more realistic views.

We developed the interface which uses the property of a global level. The same idea may be able to be applied to a local level and we are planning to research about this.

References

[1] H. Benko, A. D. Wilson, R. Balakrishnan, Sphere: Multi-Touch Interactions on a Spherical Display, In Proc. UIST 2008, pp. 77-86, 2008.

[2] R. Carroll, M. Agrawala, A. Agrawala, Optimizing Content-Preserving Projections for Wide-Angle Images, ACM Transactions on Graphics, Vol. 28 No. 3, 43:1-43:9, 2009.

[3] R. Companje, N. Dijk, H. Hogenbirk, D. Mast, Globe4D, time-traveling with an interactive four-dimensional globe, In Proceedings of the 14th annual ACM international conference on Multimedia, pp. 959-960, 2006.

[4] N. Elmqvist, N. Henry, Y. Riche, J.-D. Fekete, Melange: Space Folding for Multi-Focus Interaction. In Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems, pp. 1333-1342, 2008.

[5] G. W. Furnas, Generalized Fisheye Views, In Proceedings of the ACM CHI'86 Conference on Human Factors in Computer Systems, pp. 116-123, 1986.

[6] G. W. Furnas, A Fisheye Follow-up: Further Reflections on Focus+Context, In Proceedings of the ACM CHI 2006 Conference on Human Factors in Computing Systems, pp. 999-1008, 2006.

[7] T. Igarashi, T. Moscovich, J. F. Hughes, As-Rigid-As-Possible Shape Manipulation, ACM Transaction on Graphics, Vol. 24, No. 3, pp. 1134-1141, 2005.

[8] B. Jenny, T. Patterson, L. Hurni, Flex Projector-Interactive software for designing world map projections, Cartographic Perspectives, 59, pp. 12-27, 2008.

[9] B. Jenny, T. Patterson, L. Hurni, Graphical design of world map projections, International Journal of Geographical Information Science, Vol. 24, No. 11, pp. 1687-1702, 2010.

[10] B. Jenny, Adaptive Composite Map Projections, IEEE Transactions on Visualization and Computer Graphics, Vol. 18, No. 12, pp. 2575-2582, 2012.

[11] B. Jenny, T. Patterson, Blending world map projections with Flex Projector, Cartography and Geographic Information Science, Vol. 40, No. 4, 2013.

[12] B. Lévy, S. Petitjean, N. Ray, J. Maillot, Least Squares Conformal Maps for Automatic Texture Atlas Generation, ACM Transactions on Graphics, Vol. 21, No. 3, pp. 362-371, 2002.

[13] K. Matsuyama, K. Konno, MapSlider: A Property Based Interface for World Map Software, NICOGRAPH International 2014, 2014.

[14] K. Matsuyama, M. Okamoto, MIKAN GLOBE MT: An Interactive World Map featuring Seamless Map Projection and Guaranteed Visibility, The Journal of the Society for Art and Science, Vol. 8, No. 3, pp. 130-141, 2009.

[15] J. Nocedal, S. J. Wright, Numerical Optimization, and ed. Springer, 2006.

[16] D. F. Reilly, K. M. Inkpen, Map morphing: making sense of incongruent maps, GI '04 Proceedings of the 2004 Graphics Interface Conference, pp. 231-238, 2004.

[17] M. Sarkar, S. S. Snibbe, O. J. Tversky, S. P. Reiss, 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, pp. 81-91, 1993.

[18] S. Schaefer, T. McPhail, J. Warren, Image Deformation Using Moving Least Squares, ACM Transaction on Graphics, Vol. 25, No. 3, pp. 533-540, 2006.

[19] J. P. Snyder, Computer-assisted map projection research, U.S. Geological Survey Bulletin 1629. U.S. Gov. Printing Office, 1985.

[20] J. P. Snyder, Map Projections - A Working Manual, U.S. Geological Survey Prof. Paper 1395. U.S. Gov. Printing Office, 1987.

[21] J. P. Snyder, Flattening the Earth, two thousand years of map projections, University of Chicago Press, ISBN 0-226-76746-9, 1993.

Katsutsugu Matsuyama

Katsutsugu Matsuyama is currently an assistant professor at Iwate University. His research interests include computer graphics, information visualization and interactive systems. He received BE, ME, DE degrees in computer science from Iwate University in 1999, 2001 and 2005, respectively. He was a research associate at Future University-Hakodate from 2005 to 2011.

Kouichi Konno

Kouichi Konno is a professor of Faculty of Engineering at Iwate University. He received a BS in Information Science in 1985 from the University of Tsukuba. He earned his Dr.Eng. in precision machinery engineering from the University of Tokyo in 1996. He joined the solid modeling project at RICOH from 1985 to 1999, and the XVL project at Lattice Technology in 2000. He worked on an associate professor of Faculty of Engineering at Iwate University from 2001 to 2009. His research interests include virtual reality, geometric modeling, 3D measurement systems, and computer graphics. He is a member of IEEE.