# Focus+Glue+Context: An Improved Fisheye Approach for Web Map Services

Daisuke Yamamoto
Nagoya Institute of Technlogy
Gokiso-cho, Showa-ku,
Nagoya, Aichi, Japan
daisuke@nitech.ac.jp

Shotaro Ozeki
Nagoya Institute of Technlogy
Gokiso-cho, Showa-ku,
Nagoya, Aichi, Japan
ozeki@moss.elcom.nitech.ac.jp

Naohisa Takahashi
Nagoya Institute of Technlogy
Gokiso-cho, Showa-ku,
Nagoya, Aichi, Japan
naohisa@nitech.ac.jp

## ABSTRACT

This paper proposes a method for generating a Focus+Glue +Context map for Web map services by improving existing fisheye views methods for cartographic data. While many studies have focused on fisheye views, the problems of the excessively large distortions that maps have over their entire area and/or the high density of roads in the borders of the maps have yet to be resolved. To completely remove the distortions in both the Focus and the Context areas, we propose the inclusion of a Glue area between the two areas. Moreover, we propose the following features. (1) We propose the Dynamic Displacement function to generate a Focus and a Glue having an arbitrary convex or star-shaped polygon. (2) Since the Glue area absorbs all distortion from the Focus and the Context areas, the density of roads along the angular direction of the Glue will be excessively high. To reduce the density of roads, we propose the Following Paths algorithm to display only those roads that are along the radial direction and that are reached from the Focus area to the Context area. This method enables users to understand a map easily with high visibility of the Glue area. (3) To realize Web map services, we have to generate a map quickly. Therefore, the Glue area is generated dynamically, while the Focus and the Context areas display previously generated static maps. These maps are composited in a client Web browser by the Dynamic Map Composition method. We developed a prototype of our system that implements these features and evaluated the advantages of the proposed method. The new system can contribute to novel Web map services with fisheye views.

## Categories and Subject Descriptors

H. Information Systems [**H.5 Information Interfaces and Presentations**]: User Interfaces

## General Terms

Algorithm, Human Factors



## Keywords

fisheye views, Focus+Glue+Context, Web map service, GIS

## 1. INTRODUCTION

In recent years, advanced Web Map Services such as Google Maps [6] and Yahoo! Maps [20] have become available online. In addition, GPS-equipped mobile terminals have enabled novel mobile services such as pedestrian navigation[1].

On the other hand, when users want to search multiple areas by using existing Web digital maps, a problem that they encounter is that it becomes necessary to zoom and scroll through the maps repeatedly. For instance, consider a situation in which a user wants to confirm a route passing through multiple destinations. Although he/she has to view the wide-area map to view the route and the geographical relations between the destinations, he/she also has to view the detailed map to understand details about the target areas and intersecting routes by switching the scale of the maps.

Previous studies [12, 15, 16, 9] have proposed Focus+ Context type fisheye views that show both the detailed map (*Focus*) and the wide-area map (*Context*) in one map to solve the abovementioned problem. This method enables users to simultaneously view both maps by zooming in a manner similar to a fisheye lens. Although the fisheye view method is easy to use [11], it suffers from problems in that the entire area of the map is distorted and/or the density of roads in the corners of the map is large. Since existing fisheye views methods generate the entire map dynamically by using a displacement function, the calculation cost is high. Therefore, it is difficult to apply these methods to Web map services that need to be generated at high speed.

In our previous study [18], we have proposed the concept of the *Focus+Glue+Context* (we call it *F+G+C* in this paper) map that improves upon existing fisheye views for cartographic data. Unlike existing fisheye views, the F+G+C map has the *Glue* area between the Focus and the Context areas to remove distortions from both areas completely.

In this paper, we propose an effective method for generating high-quality F+G+C maps that are suitable for use in Web map services. Our method suitably solves problems related to F+G+C maps; in addition, it can generate maps at a high speed that is suitable for Web map services. We have confirmed the advantages of our method by developing a prototype system and performing experiments using it.

This paper has seven sections. Section 2 describes the features of our method and the problems associated with it. Section 3 describes the *Dynamic Displace function*, the *Following Paths algorithm*, and the *Dynamic Map Compo-*

*sition method*. Section 4 describes the prototype system. Section 5 presents the evaluation results of the calculation cost, accuracy, and usability. Section 6 describes related works. Section 7 presents the conclusions of this paper.

# 2. FOCUS+GLUE+CONTEXT

In this section, we describe a model of the F+G+C map and problems related to the F+G+C map.

## 2.1 Focus+Glue+Context Map

An F+G+C map has a Focus, a Glue, and a Context; these are based on cognitive maps [7, 14, 19]. As shown in Figure 1, the Focus is an area of a large-scale map that enables users to understand details about the focused area, the Context is an area of a small-scale map that enables users to understand geographical relations, and the Glue shows the routes that connect the Focus with the Context. Unlike existing fisheye views [15], in the F+G+C map, the Focus and Context have no distortion because the Glue contains all the distortion.

Let us compare existing methods (Figure 2-a, 2-b) with the F+G+C method (Figure 2-c). Figure 2-a shows a map transformed from a 1:25000 map like an optical fisheye lens [12]. Although this map enables users to view both the details of the center area and the wide-area map, it suffers from problems in that the entire area of the map is distorted and the density of roads in the corner is too high. Figure 2-b shows the map generated by improving the flat lens fisheye method [9, 11]. Since the corner of the Focus has distortion, the Context and the center of the Focus have no distortion. The Focus is based on 1:25000 map data, and the Context is based on 1:70000 map data. In contrast to Figure 2-a, both the center in the Focus and the Context are displayed based on a proper scale. Since the substantial scale of the border between the Context and the Focus is too small, however, there is a problem at the border in that many roads are overlapped. To solve these problems, we propose the F+G+C map, as shown in Figure 2-c. In contrast to Figure 2-b, the Glue between the Focus and the Context can control distortions properly.

In addition, in the F+G+C map, both the direction and the distance between the centers in any Focus and any point in the Context are accurate. If the target object is centered in the Focus, we can understand geographical relations between the Focus and the Context directly. By displaying multiple Focuses in the Context, we can generate a map that shows both the details of targets and the routes between these targets, as shown in Figure 1. In addition, since the Focus and the Context have no distortion, the F+G+C map has an advantage in that these areas can be used to display high-quality maps created by humans directly.

## 2.2 Problems in Web-based F+G+C Map

We state the problems that need to be solved to realize the use of F+G+C maps in Web map services.

### *2.2.1 Problem of Arbitrary Shape of the Focus and the Glue*

In many existing fisheye view methods for maps, the shape of the Focus is limited to a square or a circle. However, since the shape of the target area is not necessarily fixed, we require a method that supports a Focus and a Glue having an arbitrary shape such as an oval or a polygon.

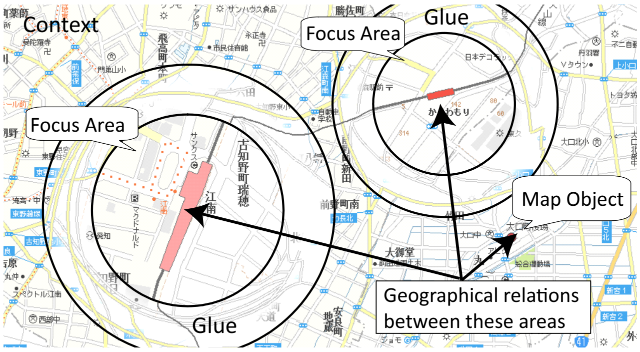


**Figure 1: Focus+Context+Glue map.**

Although the variable-scale map method based on grids [8] or the graph transformation method according to the shape of any polygon [4] can generate a Focus having a complex shape, these methods allow a distortion over the entire map without the Focus. However, since we want to generate a Glue having an arbitrary shape and avoid the distortion of the Focus and the Context, we cannot apply these methods to the F+G+C map as is. In other words, we need to generate a Focus and a Glue having an arbitrary shape without causing distortions in both the Focus and the Context. Therefore, this paper proposes the Dynamic Displacement function based on the shape of the Focus and the Glue by improving Sarkar's study [15] and Formella's study [4].

### *2.2.2 Problem of Generalization in the Glue*

Since the Glue absorbs all distortions from the Focus and the Context, its substantial scale is smaller than that of the Focus and the Context. In particular, the substantial scale along the radial direction is smaller than that along the angular direction, as shown in Figure 3. As a result, roads along the angular direction are overlapped. Since the Glue has strong distortion, its width should ideally be as small as possible. However, when the width of the Glue is small, this tendency becomes significant.

On the other hand, usual maps select displayed roads based on their degree of importance in each scale. For instance, although main roads suited to a 1:70000 scale are only displayed in the Context, narrow roads suited to a 1:25000 scale are displayed in the Focus. However, if narrow roads suited to 1:25000 maps are displayed in the Glue as well as in the Focus, the density of roads along the radial direction increases, as shown in Figure 4-a. If main roads suited to a 1:200000 map are drawn in the Glue according to the scale along the angular direction, the roads reached from the Focus to the Context cannot necessarily be displayed, as shown in Figure 4-b. When users use the F+G+C map to confirm routes, the roads reached from the Focus to the Context should be drawn in the Glue. In other words, the Glue should be drawn based on not only the degree of importance of roads but also the connectivity between the Focus and the Context. In addition, the calculation cost should be small for applications in Web map services.

To display the routes reached from the Focus to the Context in the Glue, we discuss the Shortest Path algorithm and the Following Path algorithm in this paper.

The Shortest Path algorithm is used to select the shortest path reached from roads in the Focus to the Context using

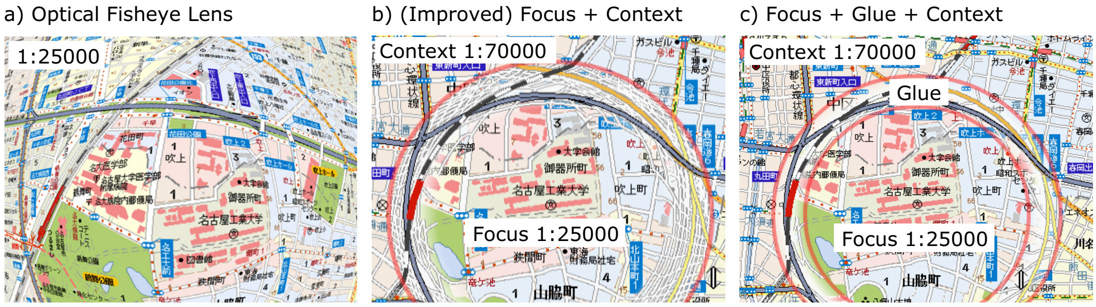


**Figure 2: Differences among fisheye view methods. a) The entire area is distorted and the corner is dense. b) The center of the Focus and the Context have no distortion. The border of the Focus is dense. c) Proposed method. The roads in the Glue are reduced by using the Following Path algorithm.**

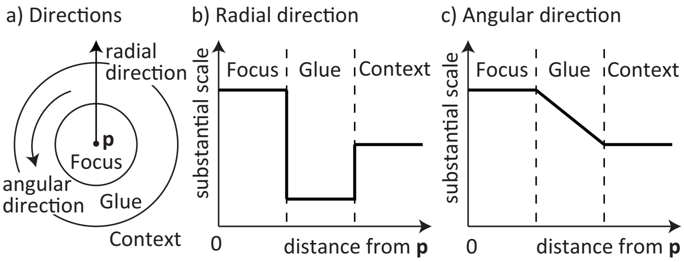


**Figure 3: Scale problems in the Glue. b) Scale along the radial direction. c) Scale along the angular direction.**

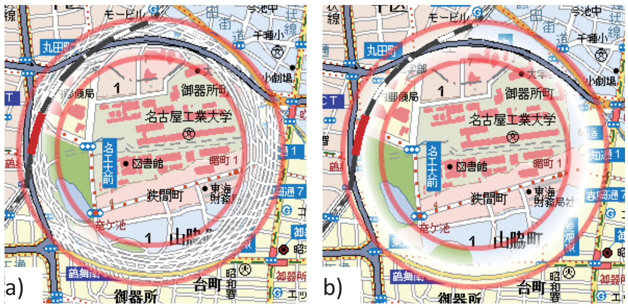


**Figure 4: a) Drawing all roads. The density of roads along the angular direction of the Glue is excessively high. b) Drawing only main roads. Roads reached from the Focus to the Context are not displayed.**

Dijkstra's algorithm [2]. Although this method can display the routes reached from the Focus to the Context, these routes may be complex and may include many intersections. Since the Glue is narrow, these complex routes may not be displayed.

Therefore, we propose the Following Path algorithm. This algorithm is used to select the following paths reached from roads in the Focus to the Context. The *following path* is the route that a human follows along a road without turning even if he/she reaches intersections. Since the following path is simple to draw, it has an advantage in that intersections and routing assistance information need not be displayed. Since this algorithm can determine roads sequentially, it is also simpler and faster than the Shortest Path algorithm. Since there is a problem in that this algorithm fails when the following path reaches a T intersection or a dead-end street, however, we have to solve this problem.

Since it is necessary to prioritize the visibility of the Glue and improve the processing speed, we adopt the Following Path algorithm to solve the abovementioned problems.

In addition, the borders of the Glue have a problem in that straight roads are bent in a manner similar to the refraction of light, as shown in Figure 8-b. Therefore, to connect these areas more smoothly, we have to consider the Dynamic Displacement function.

#### *2.2.3 Speed-up Techniques for Web Map Services*

Since existing fisheye views generate the entire map on the screen dynamically, they have a problem in that is difficult to display high-quality maps and speed-up their display. However, these are practical requirements that are essential for Web map services. In order to reduce the calculation costs, reducing the number of roads being displayed is generally effective. Although this can be achieved by drawing only the major roads, it becomes impossible to draw fine, high-quality maps.

Therefore, unlike existing fisheye view methods, we generate only the Glue dynamically. Both the Focus and the Context are displayed by using static maps generated beforehand. We propose the Dynamic Map Composition method to compose these maps at the client side. In addition, to draw the roads quickly in the Glue, we reduce the number of roads by using the Following Path algorithm.

## 3. PROPOSED METHOD

In this section, we describe the architecture of the proposed system and the data structure. We also propose the Dynamic Displacement function, the Following Path algorithm, and the Dynamic Map Composition method.

### 3.1 Proposed System

The proposed system adopts a client-server model, as shown in Figure 5. The Glue server can generate the Glue dynamically based on the Dynamic Displacement function and the Following Path algorithm from the vector map database. Static maps for the Focus and the Context are added to the raster map database divided by the mesh unit. Map images are loaded in the Web browser. The Web browser generates the F+G+C map from both the static mesh images (the Fo-

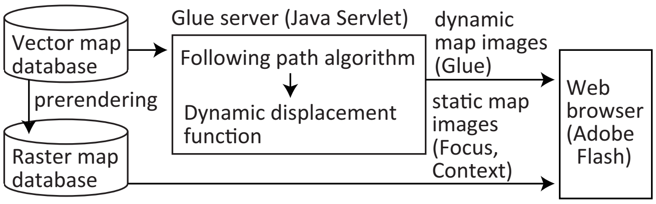


**Figure 5: Proposed system.**

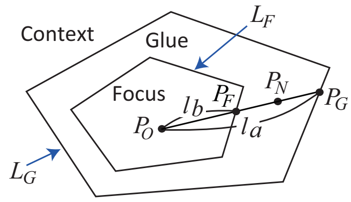


**Figure 6: Definitions of the Focus in the Context coordinates. This figure defines a pentagonal Focus.**

cus and the Context) and the dynamic images (the Glue) by using the Dynamic Map Composition method. In addition, users can control the map graphically in the Web browser.

## 3.2 Definition of the Focus Structure

First, let us define the Context coordinates and the fisheye coordinates. The Context coordinate is a normal coordinate that has no distortion. The scale of the Context coordinate is the same as that of the Context. The fisheye coordinate is a coordinate that has distortion and is transformed by the displacement function. Map objects in the Focus and Glue are transformed from the Context coordinates to the fisheye coordinates by the displacement function.

Next, let us define the shapes of the Focus and the Glue in the Context coordinates, as shown in Figure 6.

We define a polygon that shows the border between the Focus and the Glue as $L_F$. We define a polygon that shows the border between the Glue and the Context as $L_G$. $P_O$ is the fixed center point of the Focus for the displacement function. $L_G$ must include $L_F$ and $L_F$ must include $P_O$ geometrically. $L_F$ and $L_G$ must be also convex or star-shaped polygon whose whole interior is visible from a point $P_0$ without crossing any edge. $S_C$ is the scale of the Context and $S_F$ is the scale of the Focus. $M$ is the scaling factor between $S_C$ and $S_F$ and it satisfies the following expression.

$$S_F = MS_C \quad (M > 1) \tag{1}$$

For any point $P_N$ of objects in the Glue, let us draw the line that passes through the points $P_O$ and $P_N$ as shown in Figure 6. $P_F$ is the intersection point of the line and $L_F$, and $P_G$ is the intersection point of the line and $L_G$. $l_b$ is the length between $P_O$ and $P_F$, $l_a$ is the length between $P_O$ and $P_G$. $l_a$, $l_b$, and $M$ must satisfy the following expression.

$$l_a > Ml_b \tag{2}$$

The relationships between the fisheye coordinates and the Context coordinates are as follows. $l'_a$ is the length between $P_O$ and $P_G$ in the fisheye coordinates. $l'_b$ is the length between $P_O$ and $P_F$ in the fisheye coordinates. $l_a, l'_a, l_b, l'_b$, and $M$ satisfy the following expression.

$$l'_b = Ml_b, \quad l'_a = l_a \tag{3}$$

## 3.3 Dynamic Displacement Function

In this section, we propose the Dynamic Displacement function to generate a Focus and Glue having arbitrary shapes. To transform Context coordinates to the fisheye coordinates, this method generates the displacement function $DT(r)$ dynamically from $L_F$ and $L_G$ for each $P_N$.

### *3.3.1 Displacement Function*

A displacement function transforms vector data according to the distance between each node and the fixed center point.

Let us define the vector of $P_N$ in the Context coordinates as $\mathbf{v}$. $\mathbf{f}$ is the vector of $P_O$ in the Context coordinates. $\mathbf{v}'$ is the vector of $P_N$ in the fisheye coordinates. These parameters satisfy the following expression by using the displacement function $T(r)$.

$$\mathbf{v}' = T(r)\frac{\mathbf{v}-\mathbf{f}}{|\mathbf{v}-\mathbf{f}|} + \mathbf{f} \tag{4}$$

where $r$ is the distance between $\mathbf{f}$ and $\mathbf{v}$. This function enables each point $P_N$ to transform according to the distance $r$. When $r'$ is the distance between $\mathbf{v}'$ and $\mathbf{f}$, the following expression is obtained.

$$r' = T(r) \tag{5}$$

We can transform vector data using this function. Many fisheye view methods use the displacement function. In fact, Sarkar [15] defined the displacement function as follows.

$$T(x) = \frac{(d+1)x}{dx+1} \quad (0 < x < 1, d > 0) \tag{6}$$

Although one displacement function can transform any point in the circular Focus, it cannot transform all points in a complex-shaped Focus. Therefore, to use a Focus and Glue having any shape, it is necessary to modify the displacement function dynamically according to the shape of the Focus.

### *3.3.2 Requirements for Displacement Function in Focus+Context+Glue Map*

Before we propose the Dynamic Displacement function, we describe the restrictions and characteristics of the displacement function $T$ for fisheye views and the requirements of the displacement function $T$ for the F+G+C map.

In order to draw continuous maps, the displacement function $T$ must satisfy the following constraints.

- $T(0) = 0$
- $T(r)$ is a continuous function.
- $dT/dr$ is constantly 0 or more.

In addition, the displacement function $T(r)$ has the following characteristics.

- When $T(r) = ar$ is formed in the range of $r1 < r < r2$, a transformed map is $a$ times bigger than and homothetic to the original map in the range of $r1 < r < r2$. Here, $a, r1, r2$ are constant.

- When $T(r) = r$ is formed in the range of $r1 < r < r2$, a transformed map is the same as the original map in the range of $r1 < r < r2$. Here, $a, r1, r2$ are constant.
- $dT/dr$ is the scaling factor of the transformed map.
- When the gradient of $T(r)$ is smooth, the map transformed by $T(r)$ is smooth.

In the F+G+C map, in order to make multiple Focuses on a Context, we have to generate the Focuses without transforming the Context. In addition, to avoid generating apparent turns from a straight road on the borders of the Glue, we have to connect roads geometrically and smoothly on the borders of the Glue. Considering these requirements and characteristics, the displacement function $T$ must satisfy the following conditions.

- In the range of the Focus ($0 < r < l_b$), the displacement function must be $T(r) = Mr$. Here, the parameter $M$ is the scaling factor of the Focus against the Context.
- In the range of the Glue ($l_b \leq r \leq l_a$), the displacement function $T(r)$ must be the curved line that connects the Focus and the Context smoothly.
- In the range of the Context ($l_a < r$), since the map does not have to be transformed, the displacement function must be $T(r) = r$.

#### 3.3.3 Generation Method of Dynamic Displacement Function

We propose the generation method of the Dynamic Displacement function based on the previous conditions.

We measure the lengths $l_a$ and $l_b$ from a line drawn from the fixed center point $P_O$ to the point $P_N$ in the Context coordinates for each point. $r$ is the distance between $P_O$ and $P_N$. We define the Dynamic Displacement function $DT(r)$ according to the following cases.

*(1) In the range of the Focus ( $0 < r < l_b$).*

When the scale of the Focus is $M$ times bigger than that of the Context, the shape of the Focus must be zoomed $M$ times from the map data in the Context coordinates. Therefore, we define the displacement function $DT(r)$ as follows.

$$DT(r) = Mr \quad (M > 1) \tag{7}$$

*(2) In the range of the Context ( $l_a < r$ ).*

Since the transformed map must be the same as the original map, we define the Dynamic Displacement function $DT(r)$ as follows.

$$DT(r) = r \tag{8}$$

*(3) In the range of the Glue ( $l_b \leq r \leq l_a$ ).*

The roads between the Context and the Focus must be connected smoothly. Since the transformed roads are connected smoothly when the displacement function is smooth, we define the displacement function $DT(r)$ in the range of the Glue by using a cubic Bezier curve. When roads, as shown in Figure 8-a, are transformed by the linear displacement function, apparent turns with angles are generated, as shown in Figure 8-b. Users may incorrectly identify these apparent turns as real turns or intersections. Roads transformed by a cubic Bezier curve have no apparent turns with angles, as shown in Figure 8-c.

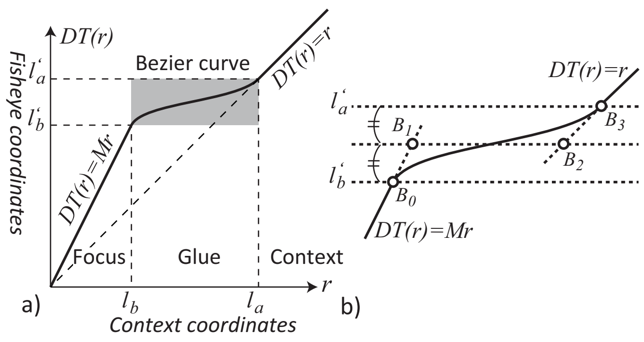


**Figure 7: a) Dynamic Displacement function in the F+G+C map. b) Bezier curve in the Glue.**

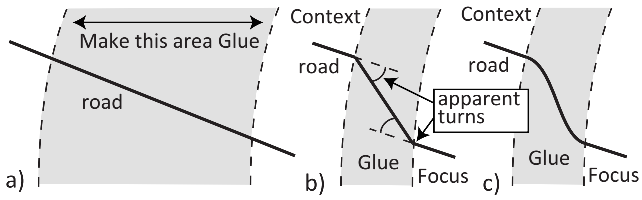


**Figure 8: a) Original road. b) Road transformed by the linear displacement function. c) Road transformed by the displacement function based on the Bezier curve.**

We define the Bezier curve by the control points $B_0$, $B_1$, $B_2$, and $B_3$, as shown in Figure 7-b. $B_2/B_3$ is the point of intersection of the line of the Focus/Context and the line $r' = (l'_a + l'_b)/2$, as follows.

$$B_0 = (l_b, Ml_b) \tag{9}$$

$$B_1 = (\frac{Ml_b + l_a}{2M}, \frac{Ml_b + l_a}{2}) \tag{10}$$

$$B_2 = (\frac{Ml_b + l_a}{2}, \frac{Ml_b + l_a}{2}) \tag{11}$$

$$B_3 = (l_a, l_a) \tag{12}$$

The function of the cubic Bezier curve is defined as follows by using these control points and parameter $t$.

$$P(t) = (1-t)^3 B_0 + 3t(1-t)^2 B_1 + 3t^2(1-t) B_2 + t^3 B_3 \tag{13}$$

Here, $0 \leq t \leq 1$. Since it is difficult to convert the function $P(t)$ including the parameter $t$ to the function $DT(r)$, we approximate $P(t)$ by dividing into lines. The displacement function $DT(r)$ is defined as follows in the range $P_x(\frac{k}{n}) \leq r < P_x(\frac{k+1}{n})$.

$$DT(r) = \frac{P_y(\frac{k+1}{n}) - P_y(\frac{k}{n})}{P_x(\frac{k+1}{n}) - P_x(\frac{k}{n})}(r - P_x(\frac{k}{n})) + P_y(\frac{k}{n}) \tag{14}$$

Here, $k$ is $0, 1, 2, ..., n-1$, $P_x(t)$ is the x-axis of $P(t)$, and $P_y(t)$ is the y-axis of $P(t)$.

Finally, the Dynamic Displacement function $DT(r)$ is defined. Then, we can generate a Focus and a Glue which have any shape that is convex or star-shaped polygon using this function.

## 3.4 Following Path Algorithm

In this section, we propose the Following Path algorithm that can be used to suitably select roads that connect from

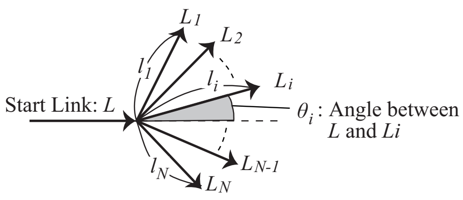


**Figure 9: Definitions of an intersection.**

the Focus to the Context. By acquiring the following paths, users can trace the routes from the Focus to the Context. Therefore, users can use the F+G+C map to search for a path search.

### 3.4.1 Criterion $R$ for the Following Path

First, let us define *the following path* mathematically. Since the following path is ambiguous in that it is defined by each individual, it is difficult to define it mathematically. Therefore, we define the criterion R whether a certain road and the road connected with it have the relation of the following path as the following rules.

We define an intersection as shown in Figure 9. The start link is $L$. Links connected with the head of $L$ are $L_1 \sim L_N$. $N$ is the number of the links connected with the head of $L$. The length of $L_i$ is $l_i$. The length $l_o$ and the angular $\alpha$ are user-defined parameters. The following four rules are applied for each link $L_i (i = 1 \sim N)$ in order.

**rule 1** When $N = 1$, $L_1$ is the following path.

**rule 2** When $N \geq 2$ and the angular $\theta_i$ between $L$ and $L_i$ is smaller than $\alpha$, link $L_i$ is the following path. Even if multiple roads are selected, we treat all selected road as the following path.

**rule 3** When $l_i \leq l_o$, we assume that the roads connected with the head of $L_i$ are connected with $L$ directly, as shown in Figure 10. When one of the roads connected with the head of $L_i$ is the following path, $L_i$ is also the following path.

**rule 4** When $N = 2$ and $L$ connects the T or Y intersection, for which the angles between $L$ and $L_1$ and between $L$ and $L_2$ are greater than $\alpha$, $L_1$ and $L_2$ are the following paths. However, since a following path is too complex if it connects many T or Y intersections, rule 4 is applied only once for each following path.

The parameters $\alpha$ and $l_o$ can control the degree of the criterion judged as the following path. Rule 3 is effective when an intersection has a slight misalignment, as shown in Figure 11. Rule 4 is effective for the connection rate of the following path, as described in Section 5.2.

### 3.4.2 Method for Acquiring the Following Path

We describe the Following Path algorithm. By repeating the selection of the following path based on criterion $R$ as follows, we can acquire the following path reached from the Focus to the Context.

For each road $L$ that crosses the edge of the Focus, apply the following algorithm.

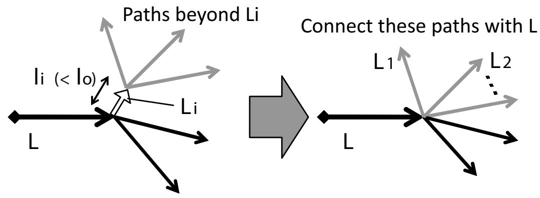


**Figure 10: Rule 3 for Criterion $R$.**


**Figure 11: Intersection having slight misalignment.**

**step 1** Add $L$ to path $F$.

**step 2** Select roads $L'$ based on criterion $R$ from the roads connected to the heads of path $F$. Add $L'$ to the heads of path $F$.

**step 3** End if the heads of path $F$ connect to the Context or $L'$ is empty. Otherwise, repeat step 2.

Path $F$ is the following path. In addition, we draw not only the following paths but also the main roads and railroads. Main roads are effective for understanding the geographical relations in the Glue and also for improving the connectivity between the Focus and the Context. Since the calculation cost of this method is not high, it is simple but effective for Web map services.

## 3.5 Dynamic Map Composition Method

The Focus and the Context in the F+G+C map have no distortion. In contrast to the Glue that is generated dynamically, the Focus and the Context can be displayed by using the static map generated beforehand.

This method is shown in Figure 12. The F+G+C map consists of the following three map layers. The lower layer is the Context, the middle layer is the Focus, and the upper layer is the Glue. The Glue layer is cut into the shape of the Glue. The Focus layer is cut into the shape of the Focus. The Focus and Context layers are displayed by loading meshed static images from the raster database and tiling the images (in a manner similar to Google Maps) [6]. The Glue layer is displayed by requesting an image from the Glue server according to user-defined operations. These mechanisms work in real-time.

# 4. PROTOTYPE SYSTEM

The vector map data adopted by the prototype system are *Standard Road Map 2007* and *Navigation Road Map 2007* obtained from Yahoo Japan Corporation. In addition, we generate static raster maps by using *ProAtlas Enterprise Server Development Kit* obtained from Yahoo Japan Corporation. Since these maps are generated from the same map data, there is no misalignment when they are overlapped. We developed the Glue server using Java Servlets and a client using Adobe Flex3.

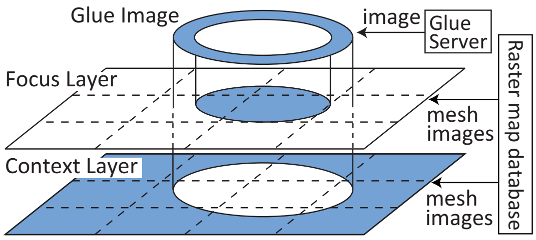


**Figure 12: Dynamic Map Composition method.**

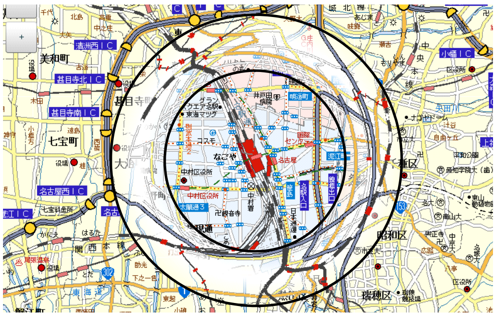

**Figure 13: Sample of the F+G+C map.**

Figure 13 shows a sample of the F+G+C map generated by the prototype system. The Focus having a scale of 1:70000 is displayed on the Context having a scale of 1:200000. The Glue is displayed between the Focus and the Context. The Focus and the Context have no distortion. The roads selected by the Following Path algorithm, main roads, and railroads are displayed in the Glue. Figure 14 shows the arbitrary shape of the Focus in the F+G+C map. Figure 1 shows multiple Focuses displayed on the Context.

In addition, we developed an interface that enables users to control the Focus interactively. By double-clicking a point in the Context, a Focus can be generated by focusing on the point. By dragging the edge of the Focus, the size of the Focus can be changed. By scrolling the mouse wheel on the Focus or the Context, their scale can be changed. By dragging the center of the Focus, the position of the Focus can be moved.

## 5. EXPERIMENTAL RESULT

In this section, we evaluate the F+G+C method from the following three viewpoints. The first is the calculation cost. The second is the availability of the Following Path algorithm. The third is the usability of our prototype system.

### 5.1 Evaluation of Calculation Costs

First, we evaluate the calculating costs of the proposed system. This system includes the following stages: (1) Obtaining vector map data from the database. (2) Selecting roads based on the Following Path algorithm. (3) Transforming roads based on the Dynamic Displacement function. (4) Drawing the Glue image. (5) Communication. (6) Displaying the map using the Dynamic Map Composition method at the client side. Although we evaluate only stages 1, 2, 3, and 4 to evaluate the server load, the other stages are also sufficiently fast.

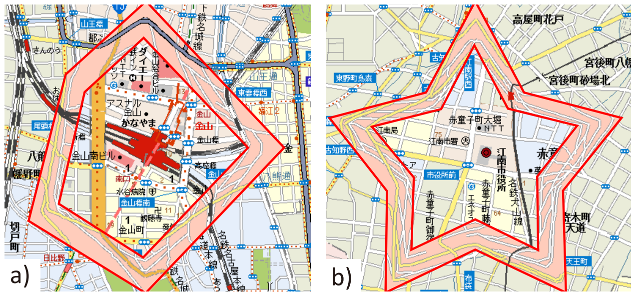


**Figure 14: a) Pentagon Focus. b) Star Focus.**

Unlike existing fisyeye view methods, the proposed system adopts the Following Path algorithm and the Dynamic Map Composition method to speed-up the display of the maps. Therefore, let us examine the calculation costs of these functions.

We compare the following methods, which are the proposed system and its subsets, based on the computation times.

**M1 method** Entire map on the screen is generated by the Dynamic Displacement function. We assume it to be the Focus+Context map.

**M2 method** Only the Glue is generated by the Dynamic Displacement function. An F+G+C map is composed by the Dynamic Map Composition method.

**M3 method** Only the Glue is generated by both the Dynamic Displacement function and the Following Path algorithm. An F+G+C map is composed by the Dynamic Map Composition method. (Proposed system)

The generated map is a square having a side of 600 pixels. The Focus is drawn at the center of the map. The Focus is shaped like a regular polygon (regular icosagon) that has an incircle having a radius of 150 pixels. The width of the Glue is 50 pixels.

In order to evaluate the effect of the number of roads, we compare the differences of scale and area. The target areas are area A (Nagoya city), which has many roads, and area B (Kakegawa city), which has a moderate number of roads. The target scales of the Context are 1:25000, 1:70000, and 1:200000. Figure 15 and Figure 16 show the result. We use the averages of 10 trials for each method.

First, let us compare M1 method with M2 method. For all scales and all areas, M2 method generated the map 2.0–3.8 times faster than M1 method. This result indicates the effectiveness of the Dynamic Map Composition method. Since M2 method generates only the Glue dynamically, unlike M1 method that generates the entire area dynamically, M2 method can reduce the database costs and drawing costs. M2 method makes use of the advantage that only the Glue in the F+G+C map has distortion.

Next, let us compare M2 method with M3 method. M3 method generated the map 1.7–3.4 times faster than M2 method. Since M3 method has to calculate the Following Path algorithm, M3 method may be considered to be slower than M2 method. Since the computation time of the Following Path algorithm is only 20% of that of all stages, however, we can ignore it. The computation time of the drawing is larger than that of the Following Path algorithm. By reducing the number of roads using the Following Path algorithm,

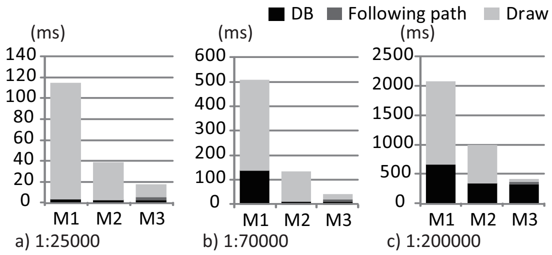


**Figure 15: Results of computation time for each method in area A. Focus is a 1:10000 map, and Contexts are 1:25000, 1:70000, and 1:200000 maps.**

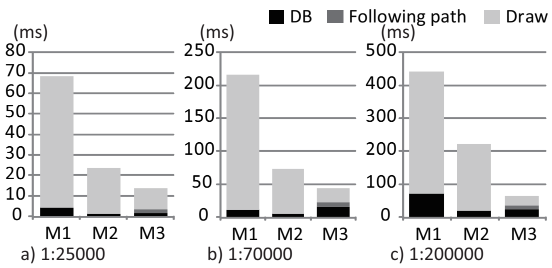


**Figure 16: Results of computation time for each method in area B. Scales are same as Figure 15.**

M3 method can generate the map faster than M2 method. Overall, M3 method is 5–12 times faster than M1 method.

In addition, we examine the influence of differences in areas and scales. Let us compare area A, as shown in Figure 15, with area B, as shown in Figure 16. Although the computation time of area A is approximately 5 times slower than that of area B and that of the 1:200000 scale map is approximately 20 times slower than that of the 1:25000 scale map, M3 method is much faster than the other methods in all areas and scales. This result shows that the proposed M3 method is effective irrespective of the area and the scale.

Finally, we examine the applicability for a Web map service. Since M3 method can generate maps having scales of 1:25000, 1:70000, and 1:20000 within 500 ms, we can provide an interactive Web map service that is suitable for this range of scales.

## 5.2 Evaluation of Following Path Algorithm

We examine the accuracy of the Following Path algorithm. Although this algorithm is fast and easy to recognize for humans, it has a problem in that all roads cannot necessarily connect from the Focus to the Context. Then, let us compare the connectivity of the Following Path algorithm (FP3) with that of its subsets (FP1, FP2). The target areas are 10 randomly selected regions.

**FP1 method** Drawing the roads selected by criterion $R$ without rule 4. This is a simple algorithm.

**FP2 method** Drawing not only FP1 roads but also the main roads in the Glue. Since main roads are connected to the Context, the paths connected to the main roads are added to the following path even if they cannot connect to the Context without main roads.

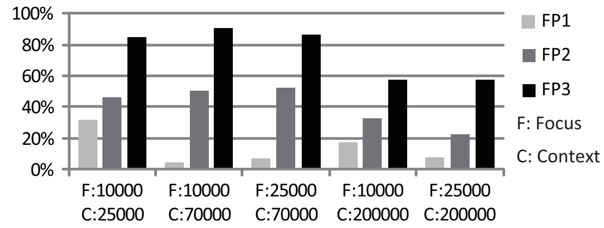


**Figure 17: Connection rates of roads connected from the Focus to the Context for each method.**

**FP3 method** Drawing the roads selected by criterion R with rule 4 and main roads. (Proposed method)

Figure 17 shows the connection rate of the roads that connect from the Focus to the Context for each method. Figure 18 shows the reduction rate of the roads reduced by each method. In the FP1 method, only 4% to 32% of roads were connected from the Focus to the Context. In particular, when the scale of the Context is small, this rate is low.

The main reason for this problem is that a following path ends when the path intersects a T intersection. In particular, when roads intersect with main roads, many of the resulting intersections are T intersections. Then, the FP2 method that solves this problem improves the connection rate (22% to 54%). However, this rate is still low. Since the Focus also displays narrow roads, the main reason for this problem is that the problem of T intersections occurs often when narrow roads starting from the Focus connect to ordinary roads. If a narrow road connects to an ordinary road as a following path once, this path may reach the Context with a high rate by the FP2 method. In fact, the FP3 method improves the connection rate significantly (58% to 91%). Since users can trace a route if some of the routes between the Context and the Focus are displayed, all road need not necessarily connect with the Context. Therefore, this rate is sufficient. In fact, there are no problems from the following questionnaires.

On the other hand, the fact that the FP3 method has to display many more roads than the FP1 method may be considered to be a problem. Since the difference of the reduction rate between the two is negligible as shown in Figure 18, however, it does not cause any problem.

In short, the Following Path algorithm (FP3 method) can increase the connection rate of roads while maintaining a high reduction rate. When the Focus is a 1:10000 map and the Context is a 1:70000 map, the FP3 method can simultaneously realize a connection rate of approximately 90% and a reduction rate of approximately 80%. Since the FP3 method reduces roads properly, it has an advantage in that we can narrow the width of the Glue with high visibility.

## 5.3 Evaluation of Usability

To evaluate the usability of the prototype system, we compared the F+G+C method with the Pan+Zoom method, which is used in Google Maps. Although we did not compare the Focus+Context method since it is not suitable for Web map services, we believe that the tendency of usability of the Focus+Context method is the same as that of the F+G+C method.

The subjects are 18 university students. The target area

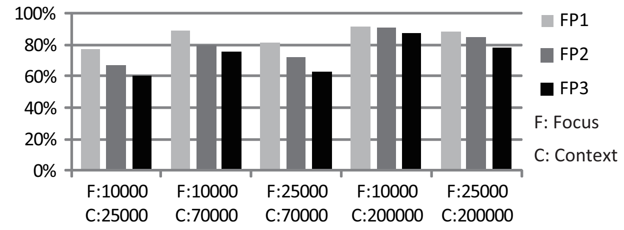


**Figure 18: Reduction rates of roads reduced by each method.**

is Sapporo city, a Japanese tourist destination. In one task, users draw a handwritten sketch map that goes through six destinations from 1 to 6 by using the prototype system, as shown in Figure 19. In addition, users have to write the name of a landmark near intersections at which a turn should be made. Since landmarks are not displayed in the wide-area map (1:70000 scale map), user have to consider not only the wide-area map but also the detailed map (1:25000 scale map) by changing scales repeatedly to understand both the geographical relations and names of landmarks. We measure the time that subjects require to complete the task by using the following two methods. Subjects practiced the system for 5 min beforehand.

**Pan+Zoom method** Users control the map by zooming and scrolling in a manner similar to that in Google Maps. Users have to switch the map between 1:70000 scale and 1:25000 scale. (Existing method)

**F+G+C method** Users control the F+G+C map by generating the Focus and scrolling the Context. Users can view landmarks without changing the scale of the map. (Proposed method)

In addition, subjects answered the following questions based on the five-point Likert scale; here, 5 is strongly agree and 1 is strongly disagree.

**Usability** Is this system easy to use?

**Continuity** Do you feel that there is continuity between the wide-area map and the detailed map?

**Search** Is it easy to search for targets?

**Relation** Is it easy to understand geographical relations?

**Lightness** Is this system speedy?

**Stress** Do you feel stressed when using this system?

First, we examine the usability of the system based on the number of operations. The F+G+C method (12.6 operations) could be controlled in fewer than half the number of operations as the Pan+Zoom method (28.3 times), as shown in Table 1, which was statistically significant at the 1% level. These results suggest that users can control the F+G+C map with less operations. Therefore, we expect that the F+G+C map can reduce the stress felt by users and also the calculation costs for servers.

Next, we examine the questionnaire results. Figure 20 shows that the F+G+C method was better than the Zoom+Pan method for all items. The items of Usability, Continuity, Search, Relation, and Stress were statistically significant at the 5% level. Since the result of the Usability for the F+G+C method was 4.89, this method can be considered to

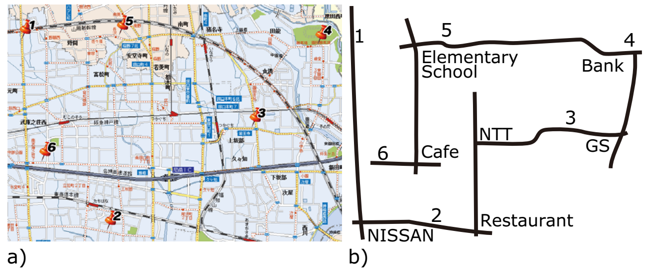


**Figure 19: An example of the task. a) A given map. b) A given sketch map example.**

**Table 1: Number of operations for scrolling by mouse-dragging, changing scales of the map, generating the Focus, and the sum of all.**

| method | scroll | scale | focus | sum |
|---|---|---|---|---|
| Pan+Zoom | 17.0 | 11.3 | – | 28.3 |
| F+G+C | 5.3 | – | 7.3 | 12.6 |

be easy to use for this task. Moreover, the result of the Continuity suggests that users can understand the geographical relations between different scale maps in the F+G+C map. We believe that this result influences the good results of the Search and Relation. Although the F+G+C method may be considered to be slower than the Zoom+Pan method, the time required to generate the Glue in the F+G+C map is lesser than the time required to change the scale in the Zoom+Pan map. In fact, in the F+G+C map, the results of the Lightness and Stress suggest that users do not feel slower than Pan+Zoom method. In short, these results suggest the high usability of the F+G+C method.

Although the F+G+C method was a novel interface for users, it has clear advantages judging from the number of operations and the questionnaire results. Therefore, we expect that these advantages will be strong more and more according to the proficiency of users.

## 6. RELATED WORK

The Focus+Context method [5] proposed by Furnas is an effective method for visualizing large-scale information. It enables a user to view an area of interest (Focus) and its overall structure (Context) together by reducing data of the Context based on the degree of interest (*DOI*). By applying the Focus+Context method to network maps, the fisheye-view method [15, 16] allows the user to zoom in to a part of a network map, like a fisheye lens. This method generates the map by using a displacement function. Harrie [12] and Gutwin [9, 11] proposed a method that displays a map with fisheye views in a small screen based on the Focus+Context method. Skopik [17] validated the usability of fisheye-view-type network maps. Unlike our method, these methods do not properly consider the problems of both the distortion and the density along the radial direction.

In addition, there have been many studies on variable scale maps. Fairbairn [3] proposed a variable scale map that progressively zooms up from the periphery to the center of the map. Guerra [8] proposed digital tourist city-maps that transform variable scale maps according to grids. Although

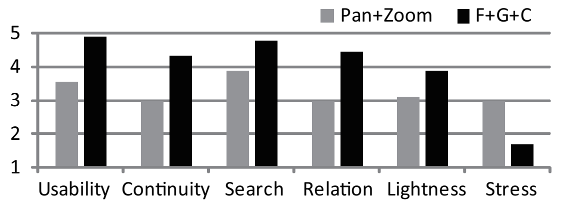


**Figure 20: Questionnaire results.**

Takahashi [18] proposed the cognitive map-based Focus operation that generates the F+G+C map by zooming the map objects such as districts and paths, and the pull operation that pulls in these objects from offscreen, he did not propose the effective generation method of the Glue.

Moreover, Lamping [13] proposed fisheye views that visualize large hierarchies by using hyperbolic geometry. Gutwin [10] applied fisheye views to a Web browser. This method zooms up on a focused area and zooms down on other areas of Web pages in a small display. Although these methods are unique, it is difficult to apply them to a map as is.

## 7. CONCLUSION

This paper proposes a novel and effective fisheye view method for generating F+G+C maps for Web map services. This F+G+C method has the following three advantages. First, the Glue controls the roads on the map properly. Since the Glue absorbs all distortions using the Dynamic Displacement function, both the Focus and the Context have no distortion. Moreover, by selecting the roads reached from the Focus to the Context using the Following Path algorithm, we can draw the Glue with high visibility. In fact, the Following Path algorithm can reduce approximately 80% of the roads in the Glue while displaying approximately 90% of the following paths reached from the Focus to the Context. Second, this method can display a Focus and a Glue having arbitary convex or star-shaped polygon. Third, this method is optimized for Web map services. The F+G+C method has higher visibility and is also up to 12 times faster than conventional methods. In addition, we have developed a prototype system and evaluated it. The results indicate that the F+G+C method has a higher usability than existing Zoom+Pan methods such as that used in Google Maps.

Moreover, we expect that the F+G+C map will find applications in tourist maps with a focus on multiple targets, guide maps with a focus on intersections, pedestrian navigation systems with a focus on the present location, and information maps with a focus on the areas that should be notified to users.

In addition, we have provided a subset of the prototype system on *the digital map technology site ALPSLAB*[1] and demo movies[2] in collaboration with Yahoo! Japan.

In future works, we have to solve the following problems. When multiple Focuses overlap in the Context, these Focuses cannot be drawn properly. We have to control the width of the Glue according to the density of the Glue. When the scale of the map is too small, we have to solve for the fact that the generation cost of the Glue is too high.

[1] http://joint.alpslab.jp/fisheye/ (Japanese)

[2] http://tk-www.elcom.nitech.ac.jp/demo/fisheye.html

**Acknowledgment** We would like to thank Yahoo! Japan Corporation for supporting us in the development of the prototype system. This work was also supported by JSPS KAKENHI 20509003.

## 8. REFERENCES

[1] M. Arikawa, S. Konomi, and K. Onishi. Navitime: Supporting pedestrian navigation in the real world. *IEEE Pervasive Computing*, 6(3):21–29, 2007.

[2] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271,1959.

[3] D. Fairvairn and G. Taylor. Developing a variable-scale map projection for urban areas. *Computers and Geosciences*, 21(9):1053–1064, 1995.

[4] A. Formella and J. Keller. Generalized Fisheye Views of Graphs. *In Proceedings Graph Drawing '95*, LNCS 1027, pages 242–253 ,1995.

[5] G. W. Furnas. Generalized fisheye views. In *Proc. of the SIGCHI '86*, pages 16–23, 1986.

[6] Google Maps, 2009. http://map.google.com/.

[7] P. Gould and R. White. *Mental Maps*. Penguin Books Ltd, Harmondsworth, Middlesex, England, 1974.

[8] F. Guerra and C. Boutoura. An electronic lens on digital tourist city-maps. In *Proc. of the 20th Int'l Cartographic Conference*, pages 1151–1157, 2001.

[9] C. Gutwin and C. Fedak. A comparison of fisheye lenses for interactive layout tasks. In *Proc. of the Graphics Interface 2004*, pages 213–220, 2004.

[10] C. Gutwin and C. Fedak. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In *Proc. of the Graphics Interface 2004*, pages 145–152, 2004.

[11] C. Gutwin and A. Skopik. Fisheye views are good for large steering tasks. In *Proc. of the SIGCHI '03*, pages 5–10, 2003.

[12] L. Harrie, L. T. Sarjakoski, and L. Lehto. A variable-scale map for small-display cartography. In *Proc. of the Symp. on GeoSpatial Theory, Processing, and Applications*, pages 8–12, 2002.

[13] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. of the SIGCHI '95*, pages 401–408, 1995.

[14] K. Lynch. *The image of the city*. MIT Press, 1960.

[15] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. In *Proc. of the SIGCHI '92*, pages 83–91, 1992.

[16] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *Proc. of the 6th annual ACM Symp. on User interface software and technology*, pages 81–91, 1993.

[17] A. Skopik and C. Gutwin. Finding things in fisheyes: memorability in distorted spaces. In *Proc. of the Graphics Interface 2003*, pages 67–75, 2003.

[18] N. Takahashi. An elastic map system with cognitive map-based operations. *Int'l Perspectives on Maps and the Internet*, pages 73–87, 2008.

[19] E. C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948.

[20] Yahoo! Maps, 2009. http://map.yahoo.com/.