A Geometric Transformation Method for Preventing Focuses from Overlapping in a Focus+Glue+Context Map

Hiroya Mizutani, Daisuke Yamamoto, and Naohisa Takahashi
Graduate School of Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya-shi, Aichi, Japan
mizutani@moss.elcom.nitech.ac.jp, yamamoto.daisuke@nitech.ac.jp, naohisa@nitech.ac.jp

ABSTRACT

We previously proposed a Focus+Glue+Context-type Fisheye View Map system called Emma. Emma is a map system comprising areas that expand to display areas of focus (Focus), peripheral areas (Context), and areas that absorb the difference between the scales of both areas and display the roads that connect the two areas (Glue). When users explore multiple destinations using Emma, they can simultaneously view wide-area maps and detailed maps of each destination by positioning a Focus on each destination. On such a map, when users move the Focus along the route in order to confirm the route from one to another, for instance, the Focuses sometimes overlap each other. This creates a problem in that the roads between the Focus and Context areas cannot be correctly drawn. In order to solve this problem, we propose a Geometric Transformation method as a solution. This method can prevent Focuses from overlapping by transforming and uniting Focuses while considering the size and position of each Focus area. We have proved the efficiency of the Geometric Transformation method by comparing it with the Water Drop method, which we previously proposed.

KEYWORDS

Geometric, Merger, Water drops metaphor, Transformation, Focus+Glue+Context, Fisheye, Web map services, Emma, Human interface, Visualization

1 INTRODUCTION

In recent years, online digital maps such as Google Maps[1] and Yahoo! MAPS[2] have been widely used. Users of these map systems have to repeatedly scroll maps and change the scale in order to view multiple destinations. For instance, when users wish to compare destinations with other separate places, they have to repeat the operations of switching and viewing detail maps, while grasping the overall positional relationship by using wide-area maps. To address this drawback, we proposed a Focus+Glue+Context-type Fisheye View Map system called Emma[3,4]. As shown in Figure 1, Emma is a map system comprising areas that expand to display areas of focus (Focus), peripheral areas at a smaller scale than those of the Focus areas (Context), and areas that absorb the difference between the scales of both areas and display the roads that connect the two areas (Glue).

When users explore multiple destinations using Emma, they can simultaneously view wide-area maps with all the destinations and detailed maps of each destination by positioning a Focus on each destination. On such a map, when users move the Focus along the route in order to confirm the route from one to another, for instance, the Focuses sometimes overlap each other and hide some parts of the Focus and Glue areas. This creates a problem in that the roads between the Focus and Context areas cannot be correctly drawn, as shown in Figure 2.

In order to solve this problem, we propose a Geometric Transformation method as a solution. This method can prevent Focuses from overlapping by transforming and uniting Focuses while considering the size and position of each Focus area. We have proved the efficiency of the Geometric Transformation method by comparing it with the Water Drop method[5,6], which we previously proposed.

Figure 1. Focus+Glue+Context map.


Figure 2. When Focuses overlap, road-network connections cannot be correctly drawn.

2 OVERLAPPING FOCUS PROBLEM

In this section, we describe the overlapping Focus problem.

When multiple Focuses are presented in a single display, the Focuses sometimes overlap each other, hiding some parts of the Focus and Glue areas, and consequently, the roads between the Focus and Context areas cannot be correctly drawn. In Emma, because the roads in Glue areas are drawn such that they connect the edges of the roads in the Focus and Context areas, the road connections are broken in the parts in which the Focuses overlap.

The problem of overlapping Focuses is difficult to solve. For instance, one could take particular note of the shapes of the Focuses, and think that the problem can be solved by creating a Merged Focus that contains these Focuses. Such a Merged Focus, however, cannot always contain all of the areas displayed in the original Focuses, because they have a larger scale than the Context areas. Therefore, let us consider the case in which two adjoining Focuses are placed as shown in Figure 3a. Figure 3b shows the Merged Focus created by combining these two Focuses. Figure 3c, the map scale of which is the same as that of the Context area in Figure 3a, shows the areas surrounded by two precise circles and an oval as the geographical areas of the two precise circle Focuses in Figure 3a and the oval Focus in Figure 3b. According to Figure 3c, the Focus area in Figure 3b does not contain the geographical areas of the two Focuses in Figure 3a (the areas surrounded by the two precise circles). Since users place Focuses on areas that they want to focus upon to confirm detailed information, we do not want to change the geographical areas (Objective areas) that the Focuses display, which is against the user's intention. In contrast, one could take particular note of Objective areas and think that the solution is to create a large Focus that can display the areas of the two original Focuses. To this end, a Focus that is much larger than the two original Focuses has to be created, and the entire Focus might not be visible in a single display.

Then, let us consider a method that transforms Focuses so they appear dented when the Objective areas overlap slightly, and merges Focuses when the distance between the Focuses is less than a certain number L. In this method, the smaller L becomes, the smaller the Merged Focus can become. Focuses that are right before uniting (when the distance between the Focuses is L) excessively transform, and their Focus areas are difficult to see. Then, the following requirements must be met to keep Focus areas easy to see.

[Requirement 1] The Focus size retention rate (the percentage of the size of a Focus area before transformation to its size after transformation) must be same as that of the other Focus.

[Requirement 2] The Focus size retention rate must not be too low.

[Requirement 3] The Merged Focus must not be too larger than the two original Focuses.

In order to solve the overlapping Focus problem while meeting these requirements, we propose the Geometric Transformation method.

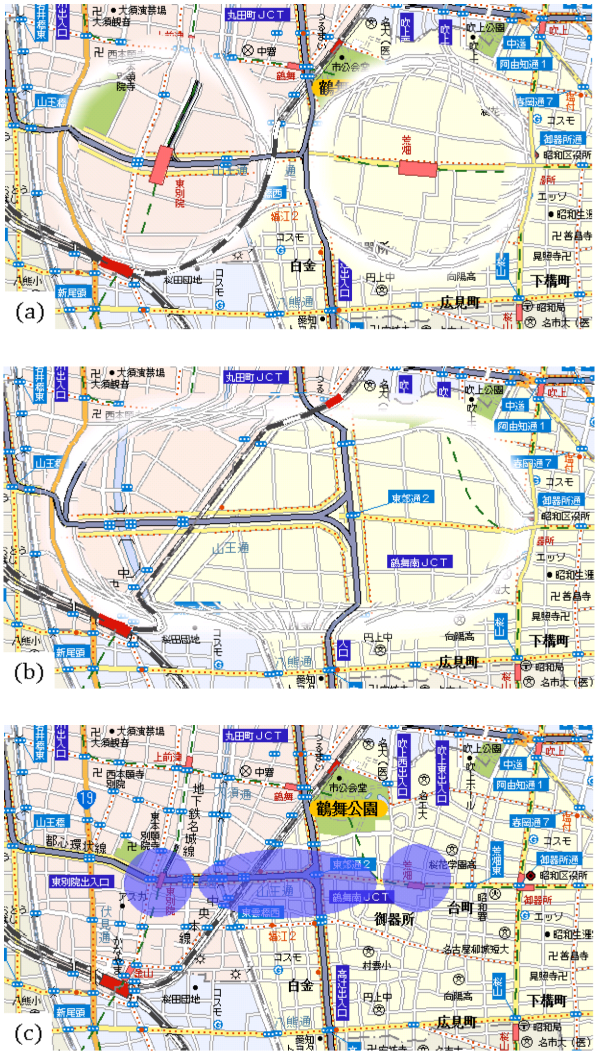

Figure 3. Relationship between the shapes of Focuses and geographical areas of the Focuses (Objective). The Focuses shown in (a) display the areas surrounded by the circles in (c), but the Focus shown in (b) only displays the area surrounded by the oval in (c).

3 PROPOSAL

In this section, we describe the proposed method. It comprises three functions.

Function1 prevents Focuses from overlapping such that they appear dented. Focuses transform according to the position and shape of each Focus. This function meets Requirement 1 by adjusting the size of the Focus area and equalizing Focus size retention rate of each Focus.

Function2 creates a Merged Focus by combining two overlapping Focuses. When a Focus size retention rate is lower than a certain number, or when the Objective areas overlap each other, this function combines the Focuses. This function meets Requirement 2 by preventing the Focus size retention rate from becoming too small. A Merged Focus display a map area that contains the geographical areas displayed by the two original Focuses right before being combined.

Function3 decreases the size of a Merged Focus by moving the Objective areas closer to each other, so this function meets Requirement 3. In Emma, normally when Focuses move, the latitude and longitude of the center points of Focus areas are the same as those of the Objective areas, but Focuses whose Focus and Objective areas with different latitudes and longitudes can be created. We call such a Focus an Off-centered Focus. Function3 makes an active Focus an Off-centered Focus. In this function, when users are moving a Focus near another Focus, the active Focus stops right before merging, and after operating the active Focus to move closer, only the map areas displayed in the active Focus moves along the movement of the mouse pointer, while the Focus area of the active Focus is stopped. In Figure 4a, the left Focus is placed such that the center points of the Focus area and Objective area are the same. When the left Focus becomes off-centered as in the situation in Figure 4a, the map image displayed in the Focus is moved to the left with the position of the Focus areas, as shown in Figure 4b. Function3 reduces the size of the Merged Focus by applying an Off-centered Focus to an active Focus. When the Focuses shown in Figure 4a are combined, the Merged Focus shown in Figure 5a is created. When the Focuses shown in Figure 4b are combined, the Merged Focus shown in Figure 5b is created. In comparing Figs. 5a and b, it appears that the size of Merged Focus can become smaller by applying an Off-centered Focus. In this function, when users move a Focus by dragging it with the mouse, Off-centered Focuses are applied according to the distance of the mouse movement. This causes users to feel like they are moving the map area displayed in the Focus, although the Focus is not moving. There is a limitation for Off-centered Focuses: Objective areas are completely enclosed by Focus areas (we will describe the details of this limitation later).

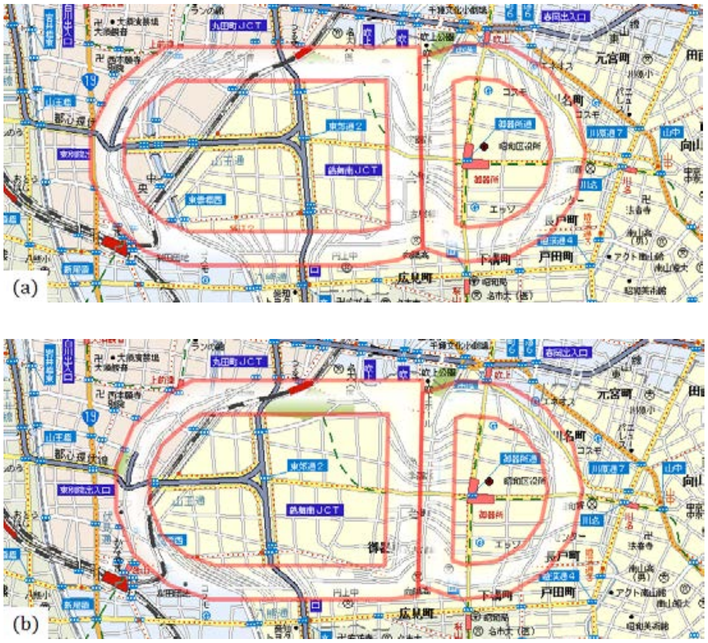

Figure 4. The two Focuses in each figure are placed at the same positions. The left Focus in (b) is applied to an Off-centered Focus, and the center points of the Focus and Objective areas of the other Focuses are same. The Objective area of the left Focus in (b) is moved to the right so that the map area displayed in the Focus area are moved to the left compared with the left Focus in (a).

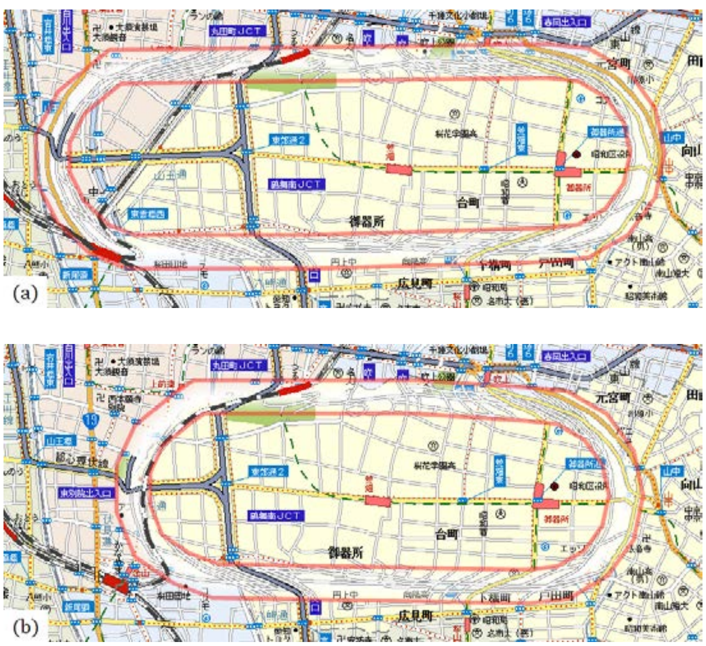

Figure 5. When the Focuses in Figs. 4a and b are combined, the Merged Focuses in (a) and (b) are created. The Merged Focus in (a) is larger than the original Focuses. However, the Merged Focus in (b) is has the same area as the original Focuses due to applying an Off-centered Focus.

4 IMPLEMENTATIONS

In this section, we describe the details of implementations of the proposed method. First, we describe the Focus definition data that are used in Emma, which contain the information that composes the Focuses. We then describe the implementation of the functions of the proposed method.

4.1 Focus definition data

In Emma, the Focus definition data set that represents the position and shape of the Focus and Glue areas. Focuses take the shape of convex, *N*-sided polygons. The coordinate type is based on XY on the scale of the Context areas. When users move Focuses, the coordinates of each data item change along with the movement. Focus data is provided in Table 1 and Figure 6. L_{OF} describes the map area of the Context area before it is expanded and displayed in the Focus areas (Objective). L_F is the border between the Focus and Context areas. L_G is the border between the Glue and Context areas. The size of L_{OF} changes with the scale of the Focus area.
Focus definition data has the following limitations: (1) L_{OF} is completely enclosed by L_F. L_F is completely enclosed by L_G. P_F is completely enclosed by L_{OF}. ($L_{OF} \subset L_F$; $L_F \subset L_G$; $P_F \subset L_{OF}$), and (2) L_{OF} and L_F are geometrically similar. ($L_{OF} \sim L_F$)

Table 1: Focus Definition Data

L_{OF}	A polygon that indicates a pre-magnified Focus area in the Context area (Objective)
L_F	A closed polygon that marks the border between the Focus and the Glue areas
L_G	A closed polygon that marks the border between the Glue and the Context areas
P_F	The center point of a Focus
$P_n(L)$	Coordinates of the n^{th} vertex of polygon L

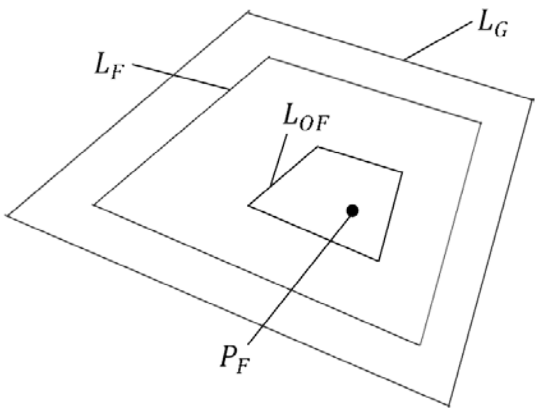

Figure 6. Focus definition data

4.2 Function1

Let us consider the case in which the borders L_G of multiple Focuses overlap, but the actual area L_{OF} does not overlap. At this point, since some parts of the Focus areas are hidden, Focus definition data L_G and L_F of each of the Focuses have to be transformed in order not to overlap, which is accomplished by using the algorithm of Function1, as described below.

We describe the algorithm in particular. Transformed shapes are calculated, dealing with two Focuses as a pair. First, when L_G of two Focuses overlap each other, a line through the intersections of the L_G is calculated (e call this line the Midline). Next, the area closer to the P_F of the other Focus outside the areas surrounded by the Midline and L_F is an area that will be removed later (we call this area D_1). In order to equalize D_1 between the two Focuses, the Midline is moved with its slope kept constant. The L_F from which D_1 is removed is the L_F after transformation. L_G after transformation is a polygon created by expanding this L_F to the Glue width. When the size of L_F after transformation is less than 60% of the size of L_F before transformation, it is determined whether an Off-centered Focus can be applied. When an Off-centered Focus cannot be applied, the two Focuses are merged by calling Function2. Function1 is called when Focuses are moved, created, or deleted.

At first glance, one might think that the roads between the Focuses are not connected when multiple Focuses are made to adjoin one another during transformation. However, if we consider the presence of extremely slight Context areas at the border L_G between the transforming Focuses, we can be assured that the roads through the Context areas are connected to one another.

The algorithm for Function1 is described below. For K Focuses, the following steps are performed for any Focus$_i$ and Focus$_j$ (i and j are integers where $i \geq 1$ and $j \leq K$).

Step 1 (Determination of overlapping) Steps 1.1 and 1.2 below determine whether the border L_G between Focus$_i$ and Focus$_j$ overlaps.

Step 1.1 This step determines whether $P_n(L_G)$ (n = 1, 2, …, N) of Focus$_i$ is in L_G of Focus$_j$. N is the number of vertices of the Focuses (both here and hereinafter). If the condition for any $P_n(L_G)$ is true, Step 2 is implemented.

Step 1.2 This step determines whether $P_n(L_G)$ (n = 1, 2, …, N) of Focus$_j$ is in L_G of Focus$_i$. If the condition for any $P_n(L_G)$ is true, Step 2 is implemented. If the condition for all $P_n(L_G)$ is false, this function is terminated.

Step 2 (Transformation process) L_G after transformation of Focus$_i$ and Focus$_j$ are, respectively, calculated using Steps 2.1 and 2.2.

Step 2.1 The Midline, which passes through the intersections of the L_G of Focus$_i$ and Focus$_j$, is calculated. Midline$_i$ is the line created by moving the Midline in the direction of the P_F of Focus$_i$ for the glue width of Focus$_i$. Midline$_j$ is the line created by moving the Midline in the direction of the P_F of Focus$_j$ for the glue width of Focus$_j$. The L_F of Focus$_i$ is divided by Midline$_i$. The L_F of Focus$_j$ is divided by Midline$_j$. In both Focuses, the area that is closer to the P_F of the other Focus of the two areas is D_1, and the other area is D_2 (Figure 7).

Step 2.2 In order to equalize D_1 of both Focuses, Midline$_i$ and Midline$_j$ are moved while their slopes are kept constant.

Step 3 (Determination of whether to call the other functions) Steps 3.1 and 3.2 below determine whether Function2 and 3 is called.

Step 3.1 Whether the condition below is met is determined. Condition: The distance between Midline$_i$ (Midline$_j$ in the event of Focus$_j$) and P_F is more than 20 pixels for both Focuses, or the size of D_2 is more than 60% of L_F for both Focuses. When this condition is met, Step 3.4 is implemented. Otherwise, the next step is implemented.

Step 3.2 Function3 is called.

Step 3.3 When the condition in Step 3.1 is met, the next step is implemented. Otherwise, Function2 is called, and Function1 is terminated.

Step 4 (Calculation of Focus shape) Steps 4.1–4.3 below calculate the shape of the Focuses after transformation.

Step 4.1 L_F after transformation is D_2.

Step 4.2 Lines parallel to each side of the polygon that comprises L_F after transformation are calculated. The lines are one glue width away from the side of the polygon.

Step 4.3 The intersections of the neighboring lines in Step 4.3 are calculated. L_G after transformation is the polygon that comprises the intersections.

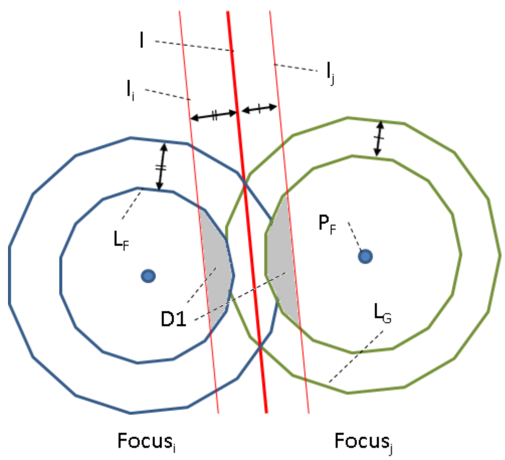

Figure 7. Method for calculating the transformed shapes in Function1.

4.3 Function2

When L_{OF} (Objective) overlap each other, or when the other Function called this function, Function2 is called. In particular, deleting two Focuses of union and creating a new Focus (Merged Focus) appears as if the Focuses unite.

We describe the algorithm of Function2 below. For K Focuses, the following steps are performed for any Focus$_i$ and Focus$_j$ (where i and j are integers with $i \geq 1$ and $j \leq K$).

Step 1 (Determination of implementing the function) When L_{OF} of both Focuses overlap, or when this function is called by Function1, the next step below is implemented. Otherwise, this function is terminated.

Step 2 (Calculation of L_{OF} of the Merged Focus) L_{OF} of the Merged Focus is the minimum convex polygon that contains L_{OF} of both Focuses. The convex polygon is calculated by applying Graham Scan[7] (convex hull).

Step 3 (Calculation of P_F of the Merged Focus) P_F of the Merged Focus is the center point of L_{OF} of the Merged Focus.

Step 4 (Calculation of the scale of the Merged Focus) The scale of the Merged Focus is the smaller scale out of the scales of the two Focuses.

Step 5 (Calculation of L_F of the Merged Focus) When the map areas of L_{OF} of the Merged Focus are displayed with the Merged Focus' scale, L_F of the Merged Focus is the polygon that contains the map, making the center point of the polygon same as P_F of the Merged Focus.

Step 6 (Calculation of L_F of the Merged Focus) L_G of the Merged Focus is calculated in a same way as Step 4.2 and Step 4.3 in Section 4.2 by using the glue width and L_F of the Merged Focus.

Step 7 (Creation of a Merged Focus) The original two Focuses are deleted, and the Merged Focus that has the Focus definition data calculated above is created.

4.4 Function3

When an active Focus is determined to combine with another Focus, applying an Off-centered Focus to the active Focus allows the map displayed in the Focus area to move with the shape while the Focus area is kept constant. In other words, the Focus area moves while the Object area does not move.

We describe the algorithm of Function3 below.

Step 1 This function is called by Function1 with Line Midline as an argument.

Step 2 On the active Focus, L_G, L_F, and P_F move in the direction from the P_F of the other Focus' to the P_F of the active Focus, such that the distance between the Midline and P_F is 20 pixels plus the Glue width. L_{OF} must be completely enclosed by L_F as described in Section 4.1.

5 EXPERIMENTAL EVALUATION and DISCUSSION

We have developed three experimental systems to evaluate the proposed method, as described below.

(1) ES1

We have developed this system based on the Water Drop method which was proposed in the previously published papers[5,6]. This system does not change the number of the vertexes of the polygon comprising a Focus. This system does not meet Requirement 1 because the size of a Focus area is not considered. However, it has a simpler data structure and its implementation is easier than that of the Geometric Transformation method due to the constant number of the vertexes. Moreover, since it combines Focuses only when the Objective areas overlap, the size of the Union Focus is same as those of the two original Focuses so this system meets Requirement 3. However, when Focuses deeply overlap and their Objective areas are right before overlapping, the Focuses are sometimes excessively transformed and the merged Focus area is difficult to see. Then, this system does not meet Requirement 2.

(2) ES2

We have implemented all the functions of the proposed method in this system. It meets Requirement 1, 2, and 3.

(3) ES3

We have implemented Function1 and 2 in this system. It meets Requirement 1 and 2, but it does not meet Requirement 3.

The difference between the three experimental systems with regards to dealing with Requirements 1, 2, and 3 described in Section 2 is shown in Table 2.

We conducted the first experiment to prove the efficiency of Function1 by comparing ES1 and ES2. We conducted the second experiment to prove the efficiency of Function3 by comparing ES2 and ES3.

Regarding the comparison between the Focus+Glue+Context map and general map systems, Yamamoto et al. (2009a) conducted an experimental evaluation to compare the Focus+Glue+Context map and Google Maps for usability. They asked subjects to work on a task and questionnaires using both maps. In the task, the subjects could quickly confirm necessary geographical information by creating Focuses instead of scrolling and zooming the map. Thus, the Focus+Glue+Context map could be controlled in fewer than half of Google Maps based on the total number of scrolling the map, changing scale, and creating a Focus. The functionality of the Focus+Glue+Context map was demonstrated by the results of the task and questionnaires.

Table 2. Each system's way of dealing with Requirement 1, 2, and 3.

	Requirement 1	Requirement 2	Requirement 3
ES1 (Water Drop method)	Not met	Not met	Met
ES2 (the proposed method)	Met	Met	Met
ES3 (the proposed method without Function3)	Met	Met	Not met

5.1 Experiment 1

We sought to clarify whether ES1 or ES2 would display map information more efficiently when Focuses transform. In particular, we prepared some types of Focuses whose shapes were different from the others. When the Focuses transformed in order to prevent overlapping, we confirmed their size. In the results, it was clarified that ES2 retains Focus areas more efficiently than ES1.

5.1.1 Experimental steps

We made 10 pairs out of 5 types of Focuses (precise circle, square, oval, triangle, large precise circle). We placed a pair of Focuses so they were overlapping and calculated their size.

5.1.2 Experimental result and discussion

Figure 8 represents the dispersion degree of Focus transformation. Hereafter, we call Focus areas removed by transformation D_1 (Figure 7). The vertical axis of Figure 8 represents the difference of the size of D_1 of the pair of Focuses divided by the summation of the size of D_1 of the pair Focuses. For instance, when the value of the vertical axis is 100%, one of the Pair Focuses unilaterally transforms, and the other one does not transform at all. For ES1, there are four pairs whose value is 100%, and the average of all pairs is 56.7%. However, for ES2, the average is 1.0%, and the maximum value is 6.5%. In the result, it was clarified that ES2 equalizes Focus size retention rate of the pair Focuses.

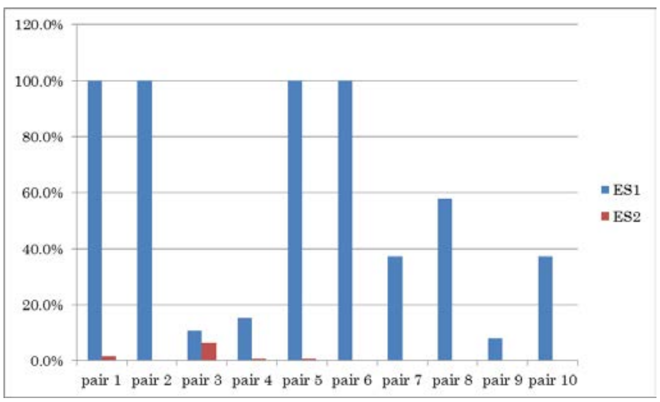

Figure 8. The dispersion degree of Focus transformation in Experiment 1.

5.2 Experiment 2

We sought to clarify the efficiency of Function3 for reducing the size of the Merged Focus. In particular, for ES2 and ES3, we calculated size of the Merged Focus and the difference between the two ESs.

5.2.1 Experimental steps

For ES2 and ES3, we performed the task twice to merge two Focuses. The first time, we moved the large precise circle Focus close to the regular precise circle Focus and merged them as shown in Figure 9a. The second time, we moved the oval Focus close to the precise circle Focus and merged them as shown in Figure 9b. We clarified the efficiency of Function3 by comparing the size of the Merged Focuses.

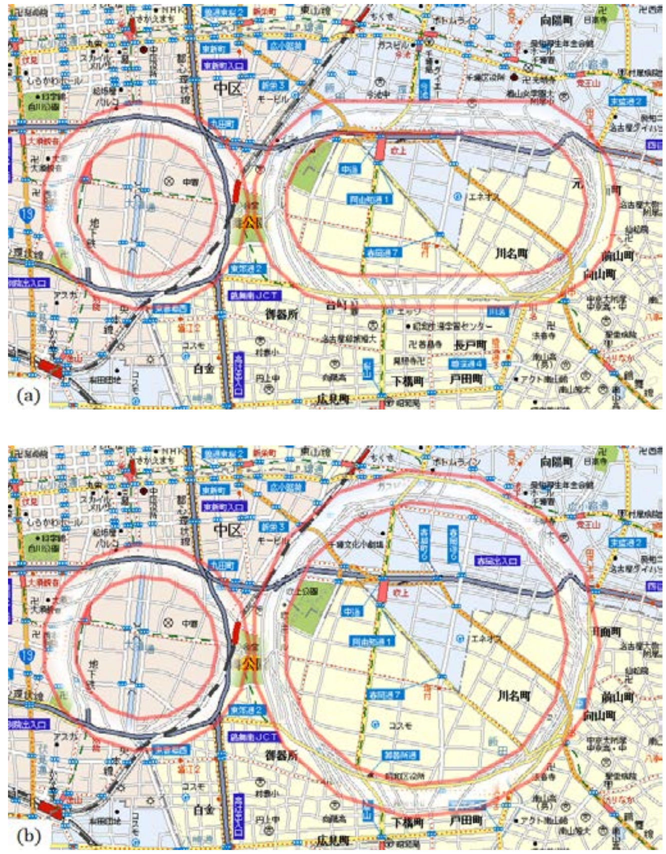
Figure 9. In (a) and (b), we moved the right Focus to the left and unite the two Focuses.

5.2.2 Experimental results and discussion

Table 3 represents the sizes of the Merged Focuses for ES2 and ES3. For both Focus pairs, it was clarified that the size of the Merged Focuses for ES2 were 80% that of the Merged Focuses for ES3. This result shows that Function3 is efficient to reduce the size of a Merged Focus.

Table 3. Merged Focus' size on ES2 and ES3 on experiment 2. Size is measured in square pixel.

	ES2	ES3	Difference	rate（ES2 / ES3）
Oval Focus and precise circle Focus	111841.6	92762.3	19079.2	82.9%
Large precise circle Focus and precise circle Focus	166725.8	134823.4	31902.4	80.9%

6 CONCLUSION

In this paper we proposed a Geometric Transformation method in order to solve the overlapping Focus problem in the Focus+Glue+Context map Emma. The proposed method prevents Focuses from overlapping by transforming and combining them while considering their sizes and positions. Moreover, it retains the areas of interest to the user.

We conducted two experimental evaluations and clarified that the proposed method can display map information efficiently and solve the overlapping Focus problem. We seek to conduct an experiment of usability for future investigation.

7 REFERENCES

[1] Google Maps, http://maps.google.co.jp

[2] Yahoo! MAPS, http://maps.yahoo.com

[3] N. Takahashi, "An Elastic Map System with Cognitive Map-based Operations," International Perspectives on Maps and the Internet. pp. 73-87, 2008.

[4] D. Yamamoto, S. Ozeki, and N. Takahashi "Focus+Glue+Context: An Improved Fisheye Approach for Web Map Services," 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 101-110 (2009).

[5] H. Mizutani, D. Yamamoto, N. Takahashi, "A Fusion of Multiple Focuses on a Focus+Glue+Context Map," Intelligent Interactive Multimedia Systems and Services, pp. 11-21 (2012).

[6] H. Mizutani, D. Yamamoto, N. Takahashi, "A preventing method for overlapping focuses in a Focus+Glue+Context Map," International Journal of Knowledge and Web Intelligence, pp. 50-69 (2013).