DEWS2004 I-11-03

A Label Placement Method for the Context Aware Map Synthesizer

Jegajothi KARTHIK† Yoshihide HOSOKAWA†† Naohisa TAKAHASHI††

† Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology Gokiso-cho, Showa-ku, Nagoya, 466–8555 Japan

†† Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology Gokiso-cho, Showa-ku, Nagoya, 466–8555 Japan

E-mail: †jayks712@moss.elcom.nitech.ac.jp, ††{hosokawa,naohisa}@elcom.nitech.ac.jp

Abstract The Elastic map synthesizer is a map projection technique where users are allowed to apply different scale transformations to different parts of the map according to the user's interest. Using the map synthesizer it is possible to blow up places of higher interest while other places are still in a smaller scale. In the map synthesizer, while map features can be viewed at different scales, some text labels either come too close to each other or overlap other labels in the neighborhood. In either case the labels hide parts of interest of the user. In this paper, we present a new label placement method for the context-aware map synthesizer. The main feature of our method is to compute user's focus areas from the transformation of a given map automatically, and apply the Tabu search and random filtering algorithms to the labels on these areas. By this strategy, our method makes it possible to put more labels without conflicts in a short time. This method improves the visibility of the area of interest of the user. We show by experimental results, the feasibiltity and the effectiveness of our method.

Keywords Context Aware map Synthesizer, Point Feature Label Placement Problem, Tabu Search

1. Motivation

There are many map projection techniques that have been proposed over the years. The Context Aware Map Synthesizer is one such technique. What makes this projection technique different from other techniques is that it allows the user to manipulate its scale at run time, allowing the user to view different parts of the map at different scales at the same time.

Nevertheless when allowing dynamic changes to be made on graphical features of the map, same rules that apply for graphical features cannot be applied on the labels that describe the graphical features.

(1) Any label present on the surface of the map should not overlap any other label on the map.

(2) The size of the labels should be such that they are easily comprehensible.

(3) For every label displayed it should be clear which graphical map feature it associates with.

The Context Aware Map Synthesizer [1] is a special kind of map projection technique, where the map frame can be manipulated as if it were an elastic sheet. With its help the user can automatically change the scale of different parts of the map. This would mean that at any point of time, a single frame of the map would be displaying various map features at various scales at the disposal of the user. The Context Aware Map Synthesizer has been developed keeping in mind web based automated map projection as one of its foremost applications.

In previous versions of the Map Synthesizer Priority selection methods were used to decide upon which labels should be displayed under certain zoom conditions. To manually set the priority of labels in each kind of configuration would become a cumbersome job considering the fact that maps are used universally.

In this paper we present a label placement method for the context aware map synthesizer. The main features of this method is that we make it possible to display more number of labels without conflicts on the map by using the Tabu Search Algorithm followed by the Random Filtering techniques. These methods are applied on a reduced area of the map.ie., the user's focus area only. Hence the Tabu Search Heuristic does not take up a lot of computation time.

The rest of the paper has been divided into three parts where we describe the Context Aware Map Synthesizer and the problem of conflicts with respect to the map synthesizer, Implementation of the proposed method with respect to the map synthesizer, related work based on the label placement problem. Before concluding we show the various results that were obtained by experiments conducted on the map under different constraints and conditions.

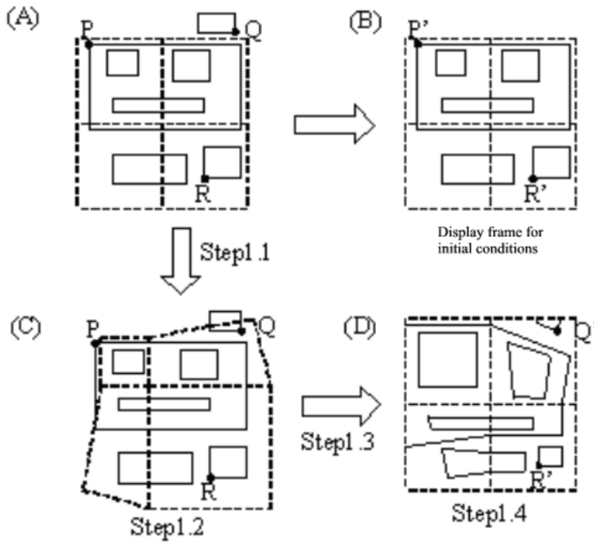


Fig 1　Transformation of map by the use of space filter

2.　The Context Aware Map Synthesizer

The Context Aware Map Synthesizer makes use of the following three data for its construction:

Map Data:

This comprises of elements and objects that are placed on the map. Picture elements like lines and symbols come under the category of elements. Objects comprise of roads and buildings that are placed on the map.

Priority of Objects:

The priority of the objects depends on the conditions provided by the user for the display of the objects.

Space Filter:

This filter can be applied to a particular space so that transformations maybe made according to the user's conditions.

The Context Aware Map Synthesizer controls and displays the map data depending on the priority specifications of the user and the changes made on the space filter. The process is described in the following 2 steps:

Step -1: Implementation of the space filter

The objects on the map are projected depending on the transformation of the coordinates of the space filter.

Step-2: Management of map objects

Selecting or deselecting objects that are to be displayed on the map depending on their priorities.

Introduction to the Space Filter

The working of the space filter is described using Figure 1. In the figure solid lines represent map objects and the dotted lines represents the frame over which the map is displayed.

Initially all vertices of all map data within the frame are projected from the map coordinate system to the display frame's coordinate system.(Figure 1(A) → Figure 1(B)). In this case the projection is made as such to the display coordinates. Points P and R become points P' and R' respectively. However point Q, does not appear in the display frame because its coordinates lie beyond the scope of the display. Every point that lies within the display frame undergoes similar transformation and are projected to the display coordinates.

However when there is a change in the display coordinates according to the proposal of the map synthesizer the transformation is done in the following way.

Step1.1: Change in the coordinates of the display frame: The map data is not transformed at this time. The dotted line in figure1[C] illustrates this.

Step1.2: Projection of the points of the map objects with respect to the new display frame. As we see in figure 1[C] points Q and R are projected to the display frame.

Step1.3: Reconstruction of the map: The map is reconstructed inclusive of all changes that were incurred in Step 1.3. For example, considering figure 1[C], the mesh on the bottom right is magnified. In order to bring it back to its original position the size of the object whose vertex is R is reduced to a smaller scale as shown in figure 1[D].

Specification of Priority of Map Data

The map synthesizer allows the following three parameters to be controlled and manipulated by the user.

Parameter 1: Display ON/OFF

Describes whether to display or not to display.

Parameter 2: Selection Criteria

Allows the user to select the threshold of magnification

Parameter 3: Magnification

Allows the user to select the scale of the map elements.

Depending on the aforementioned parameters, the priority of the elements on the map can be controlled in the following manner.

Step 2.1: Depends on Parameter 1. If it is set to OFF then the object is not displayed and if it is set to ON continues to next step.

Step 2.2: After transformation on the space filter, the magnification of the objects that lie within the display are calculated and depending on Parameter 2, the program decides whether or not to display the map feature. If the magnification lies below the threshold specified in parameter 2 the object is not displayed otherwise it continues to the next step.

Step 2.3: Depending on the magnification and the size of the element the display size of the object is decided.

Thus depending on various conditions that are provided the map synthesizer can control and restrict the number of map elements that are on display.

Explaining the use of the Map Synthesizer with an example: Consider a case where the user has to travel from Place A to Place B. Both place A and place B are accessible by train, nevertheless it is necessary to walk for quite a distance from the station. Assuming the user is unfamiliar with both place A and place B, he/she will require a detailed view of Place A and Place B and also the overall view of the train route from Place A to Place B. Under normal con-

Fig 2 The context aware map synthesizer introduces new conflicts between the labels when slight scale changes are made

ditions the user would end up with three frames each containing a detailed view of Place A, a detailed view of Place B and an overall view of the route from Place A to Place B repectively. In the case of the Context Aware Map Synthesizer all three frames would be integrated into a single frame.

With such an application as the objective, the map synthesizer cannot predict the different configurations in which the user would be viewing the map. Hence it would be impossible to previously decide the labels that are to be displayed and their positions on the map with respect to the map features. Additionally, since the scale changes in the map are irregular, possibilities of labels conflicting with each other under certain circumstances increases. For this reason it is necessary to have a method by which appropriate label positions may be calculated in real time.

Figures 2 shows model pictures of the context aware map synthesizer. Both the figures illustrate the same area of the map. The only difference between the two figures is that while the first figures contains all the features in a uniform scale the second figure has slight scale changes in the marked area.

From the figures we see that a small change in the scale of the map at certain places may move a few labels around the map and hence may trigger new conflicts among the labels. It is also possible that at times older conflicts may get corrected. Hence a need arises for real time selection and display of labels on the map.

3. Our Label Placement Method

The label placement problem that arises in the map synthesizer can be considered similar to the Point Feature Label Placement.

Definition of PFLP

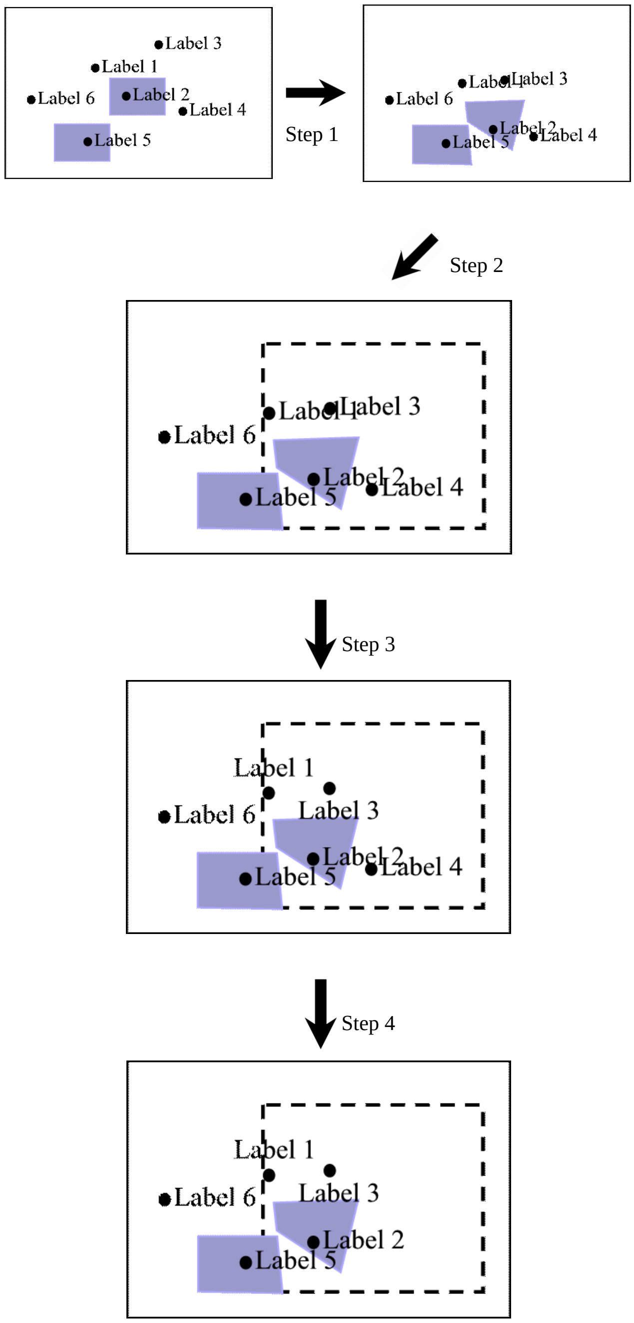
Given a set of n point features in the Euclidean plane, each feature needs to be labeled by placing a fixed text near to it. The positions allowed are restricted to a set of p places in the feature's surrounding. [2]

3.1 The Execution Procedure

The label placement method proposed in this paper is performed by the following procedures. Figure 3 shows the various steps that were followed in the implementation of the label placement method.

Step 1 : Scale transformation on the Map Synthesizer

As explained previously the Map Synthesizer allows for non-uniform scale changes on the features of the map. As shown in



Step 1. Scale transformation on the Map Synthesizer
Step 2. Selection of area of focus
Step 3. Application of Tabu Search
Step 4. Random Filtering of conflicting labels

Fig 3 Our Label Placement Method

the figure there have been cases where the scale changes affect the legibility of the labels. The exact process of scale transformation is explained in the section 2.

Step 2: Determination of user's focus area

The Context Aware Map Synthesizer allows the user to select preferred areas at run time. Therefore as a requirement it is best to give more importance to the labels that are present within the users interest. The labels on the other areas can be given secondary importance. For this reason it is necessary to calculate the users focus area on the map at run time. The exact process undertaken for the determination of the user's focus ares is explained in the section 3.

Step 3: Application of Tabu Search

When conflicts occur between labels on the map in most occasions these conflicts may be corrected by moving the labels to a different place around the point. Since the labels would still be around the point, there will not be any confusion as to which geographical feature it describes. Currently a new location for each label is chosen from nine possible locations around the point. An overview of the algorithm is explained in the following subsection.

Step 4: Random Filtering

Even though correction by the Tabu Search algorithm has proved to be useful on many occasions there are certain occasions when Tabu Search fails to correct all the possible conflicts on the map. This happens in cases where the density of labels present in the area is so high that no matter where the labels are moved to they still conflict with other labels in the area. One probable solution to this problem would be to remove certain labels in the area and display the other labels. Currently we have devised a random deletion method that removes either of the conflicting labels at random.

3. 2 The Tabu Search Algotrithm

Tabu Search is a heuristic method for solving combinatorial optimization problems. It is an iterative technique, which explores a series of feasible states by a sequence of moves. The best move in each iteration is chosen as the current move in the iteration. To avoid repeatedly returning to a previous solution, the most recent moves are classified as Tabu or illegal. This implies that in future iterations the algorithm is refrained from going back to a previous solution. The Tabu Search Algorithm in general can be written as follows:

Simple Tabu Sarch Algorithm:

Step A: Select an initial condition s. Initialize size of $Tabu - Set$ to zero. Let $x* = s$. Determine value of objective function $f(s)$.

Step B: Select a new solution s such that $f(s) < f(x*)$. Let $x* = s$.

Step C: If not stopping condition update Tabu List and Candidate List and return to Step B.

Step D: If stopping condition, s is determined to be the solution.

Objective Function: An evaluation function determined from the relationship between labels on the map.

CandidateList: A list of all labels that can be possibly moved on the surface of the map.

Tabu List: A list of labels that have been moved in the most recent iterations.

Stopping Conditions: If conflict values have been reduced to zero or to a minimum tolerant value or if the number of iterations have exceeded a minimum threshold.

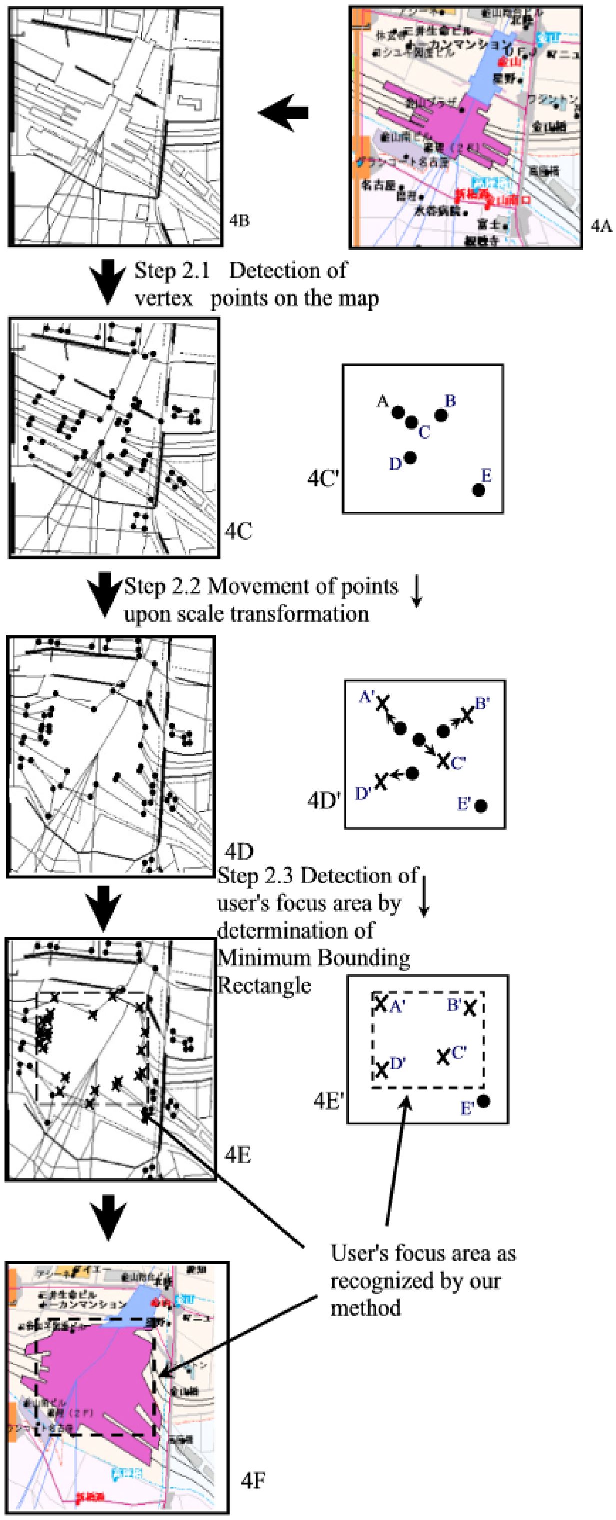


Fig 4 Determination of Focus Area

4. Determination of user's focus area

All features on a vector map are primarily defined by points on the map surface. These points form the basic building blocks for all the lines and polygons that are drawn on the map surface. Irregular scale changes on the map surface result n changes in the position of the points in the area with respect to the other points in the map. Such points that experience a change in their relative positions are calculated. The smallest rectangle that encloses all these points gives the focus area of the user. The steps followed are illustrated in Fig 4 and are explained as follows:

Step 2.1: Fig.4A shows a prototype map. Fig. 4B is the same map shown as line drawing for sake of clarity. All vertex points of the map features in Fig 4B are calculated. Fig. 4C shows all the vertex points. Let the set of points before transformation be given by,

$$P = \{p_1, p_2, ..., p_n\}$$

For easier understanding Fig $4C'$ is a simpler version of Fig. 4C. The set of points on Fig $4C'$ is given by,

$$P = \{A, B, C, D, E\}$$

Step 2.2: When scale transformations are made on the map as shown in Fig. 4D, the points on the map move with respect to each other. The new positions of the points would be given by set,

$$P' = \{p'_1, p'_2, ..., p'_n\}$$

Fig $4D'$ shows the actual movement of the points. The set of points on Fig $4D'$ is given by,

$$P' = \{A', B', C', D', E'\}$$

Step 2.3: P'' is a set of points from set P' such that the individual points change in their relative positions. On doing a one to one comparison of each of the elements in set P and P', the set of points P'' can be determined. In our example the set P'' would be given by,

$$P'' = \{A', B', C', D'\}$$

The Minimum Bounding Rectangle that encloses all the points belonging to set P'' gives the focus area of the user. Fig $4E'$ shows the the rectangle enclosing all points given by P''.Fig 4E shows the focus area shown in the original map.Fig 4F shows the original map.

5. Experimental Evaluation

We have performed an experiment to clarify the feasibility of our label placement method.

5.1 Experimental method

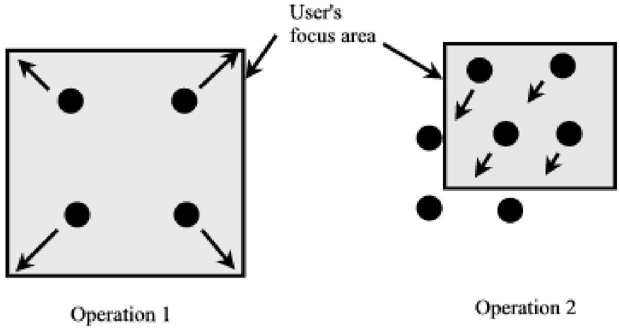


Fig 5 Determination of focus area with respect to the operations of the Contest Aware Map Synthesizer

Table 1 Analysis - magnification, focus enabled

Method (considering user's focus area)	No of Labels displayed	No of conflicting labels	Execution time (secs)
Method-1	31	26	0.321
Method-2	17	10	0.491
Method-3	18	0	0.36
Method-4	13	0	0.24
Method-5	31	7	0.501
Method-6	17	0	0.261
Method-7 (our method)	28	0	0.401
Method-8 (our method)	17	0	0.261

Table 2 Analysis - magnification, focus disabled

Method not considering uesr's focus area	No of Labels displayed	No of conflicting labels	Execution time (secs)
Method-1	382	219	0.38
Method-2	101	40	0.171
Method-3	236	0	2.293
Method-4	78	0	0.22
Method-5	382	125	3.966
Method-6	101	3	1.112
Method-7	311	0	5.689
Method-8	100	0	0.27

The context aware map synthesizer offers two charecteristic operations namely magnification and pulling in features from outside the current frame. We have applied eight label placement techiniques to each of these two characteristic operations. In each case we have evaluated and compared the number of conflicting labels and the exectuion time.

Operation 1: Magnification

When a map feature is magnified evenly, magnification occurs in the center of the rectangle that is determined as the area of focus. All the points that are determined are evenly distributed on all four sides of the rectangle.

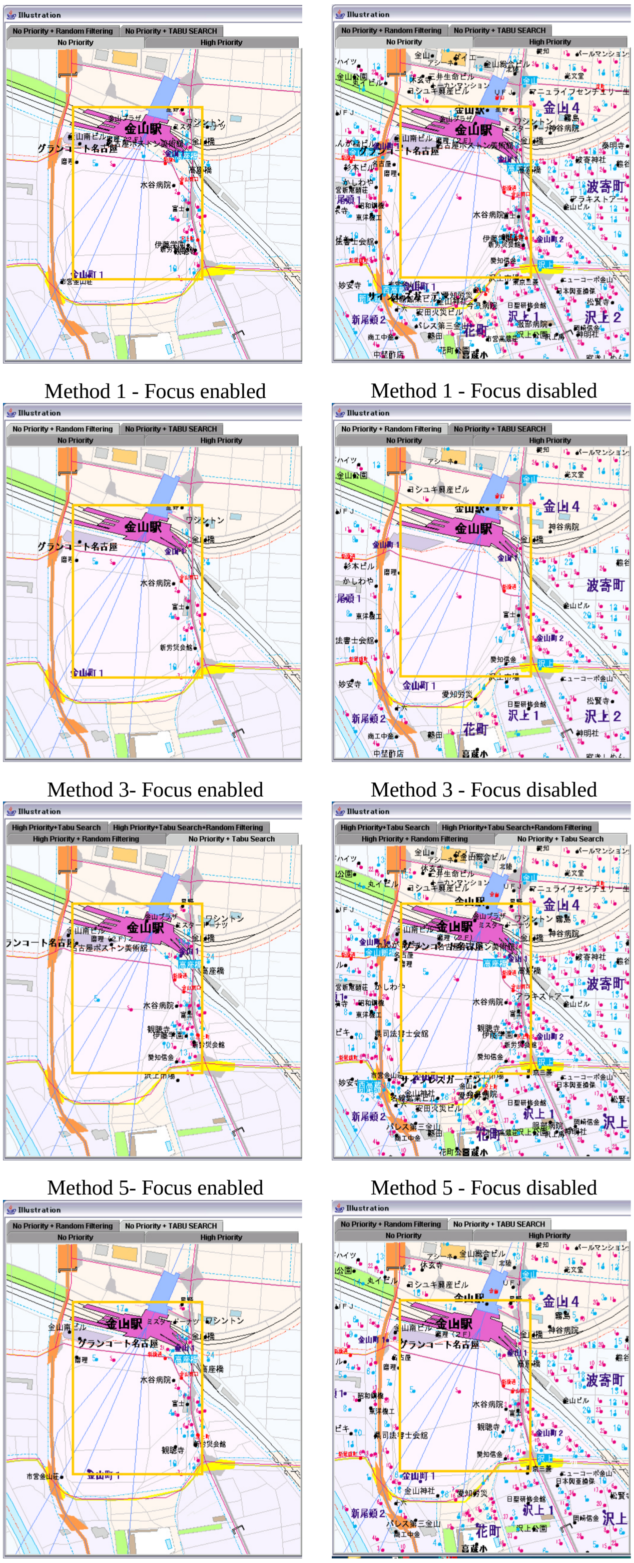
Operation 2: Pulling in features from outside the current frame

When new features that aare not currently in the frame of the map are pulled into the frame, the user's focus area would be the new features that have entered the frame. According to the process of determination of user's focus area described in this paper, the new area that has entered into the screen is correctly determined to be the user's focus area.

The eight methods that have been applied to evaluate and compare the number of conflicting labels and the execution time are explained as follows:

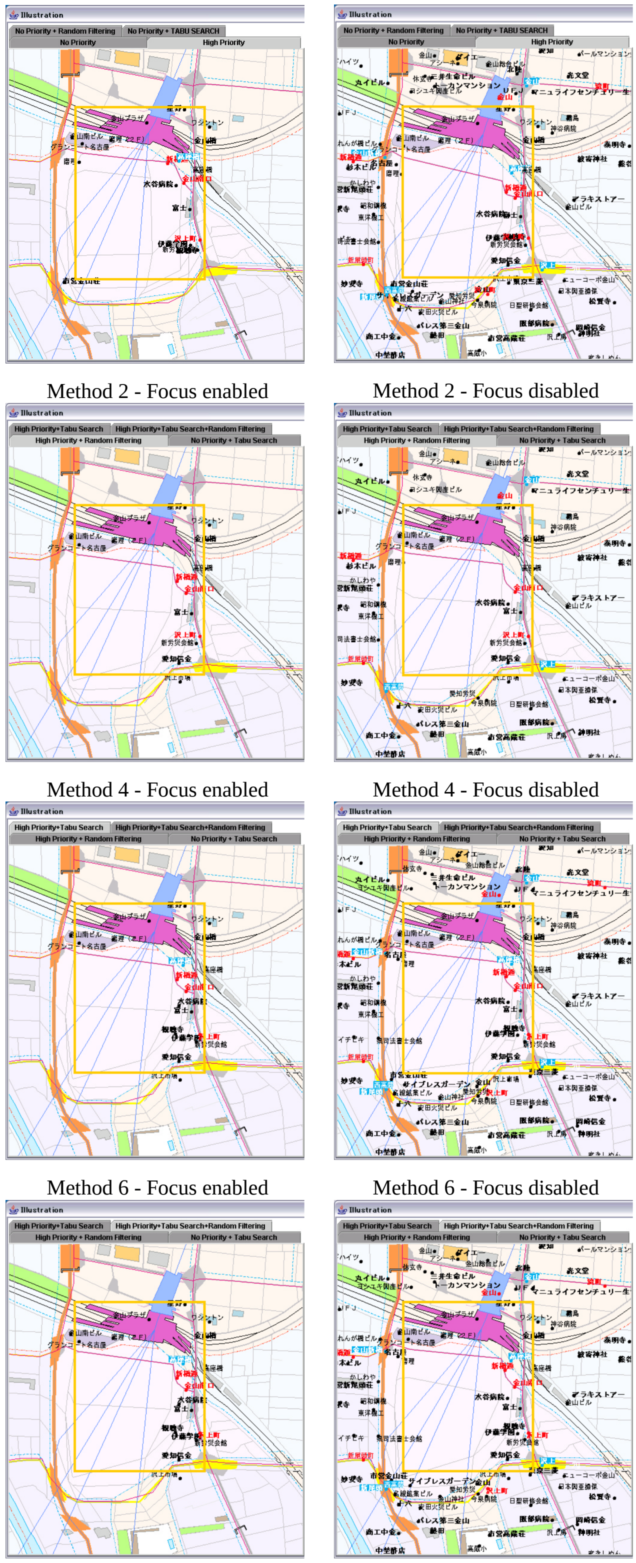
[Method-1:-No Priority]In this version of the map we have allowed all labels to be displayed without any priority.

[Method-2:-High Priority]This is the previous version of the map



Method 1 - Focus enabled
Method 1 - Focus disabled
Method 3- Focus enabled
Method 3 - Focus disabled
Method 5- Focus enabled
Method 5 - Focus disabled
Method 7- Focus enabled
Method 7 - Focus disabled

Fig 6 Map labeling - priority disabled



Method 2 - Focus enabled
Method 2 - Focus disabled
Method 4 - Focus enabled
Method 4 - Focus disabled
Method 6 - Focus enabled
Method 6 - Focus disabled
Method 8 - Focus enabled
Method 8 - Focus disabled

Fig 7 Map Labelling - priority enabled

synthesizer where labels were explicitly given priorities and hence displays a reduced number of labels.

[Method-3:-No Priority + Random Filtering]In this version all labels have been placed without priority settings but a few have been removed at random when they conflict with other labels.

[Method-4:-High Priority + Random Filtering]The random filtering technique is applied on the original version of the Context Aware Map Synthesizer.

[Method-5:-No Priority + Tabu Search]The Tabu Search algorithm has been applied on the labels present. All labels are considered without the consideration of priority.

[Method-6:-High Priority + Tabu Search]The Tabu Search algo-

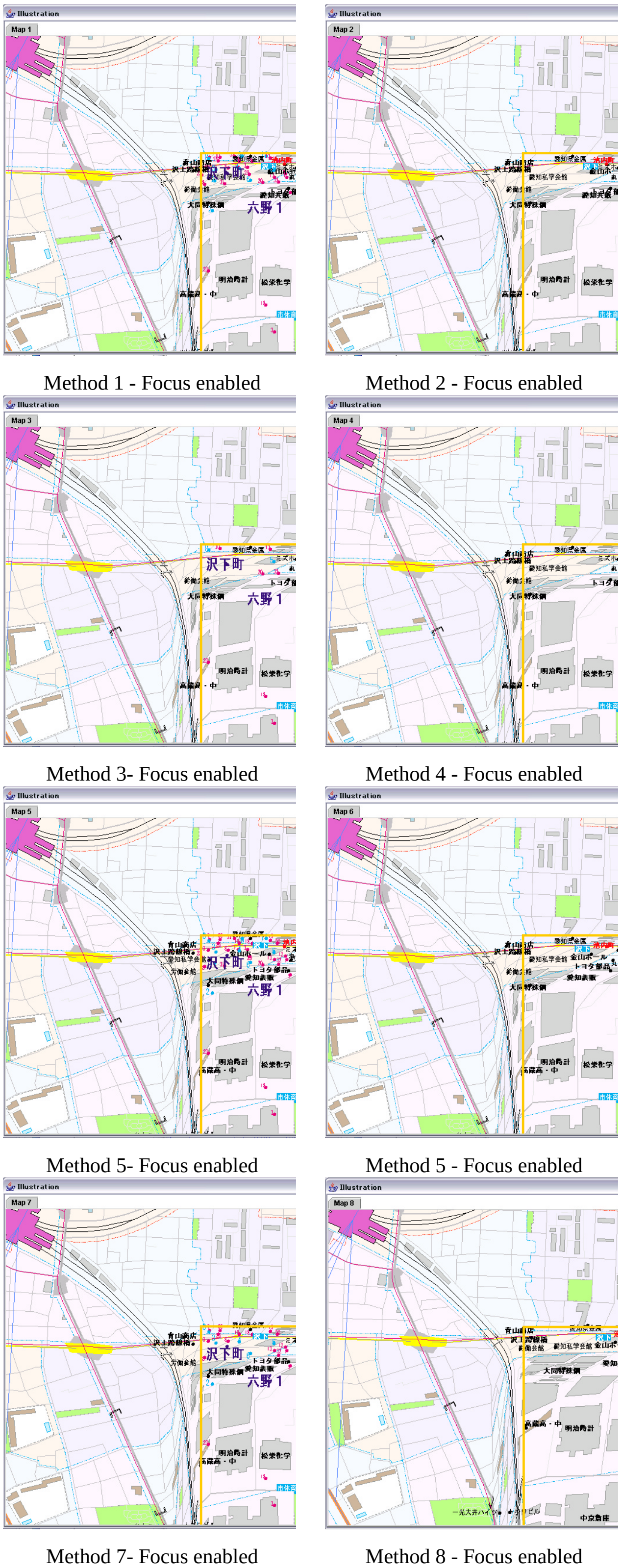


Fig 8 Map labeling - Focus on area pulled in from outside the frame

rithm is applied on the original Context Aware map Synthesizer.

[Method-7:-No Priority + Tabu Search + Random Filtering] After application of Tabu Search Random Filtering is applied to the map. Priority is not considered.

[Method-8:-High Priority + Tabu Search + Random Filtering]After application of Tabu Search to the Context Aware Map Synthesizer the Random Filtering is applied. Priority is considered.

5. 2 Experimental results and consideration

The results shown in Table 1, Table 2, Table 3 and Table 4 indicates that our method removes conflicting labels in least possible time, because the algorithm is applied to only a part of the map.ie., only at the user's focus area.

By the usage of method 7 more labels are placed on the map without conflict than methods 1, 3 and 5. And by the usage of method 8 more lables are placed on the map than methods 2, 4 and 6. Method 7 allows us to display more lables on the map surface with lesser conflicts when compared to Method 5.

On comparing the execution time of the various methods in Table 1 with Table 2 and those in Table 3 with Table 4, we see that by the determination of user's focus area we have considerably reduced the execution time, while at the same time managing to display more number of labels. Methods 7 has an execution time of 5.689 seconds and 9.974 seconds when user's focus area is not considered. While otherwise the time of execution is brought down to 0.401 secs and 1.332 seconds.

One of the main objectives of the Context Aware Map Synthesizer is that it should be usable in a web based environment. For a successful web application the execution time plays a very importatnt role. By experimental results we show that the method proposed in this paper solves the problem of label placement within minimum time. The experimental results have thus demonstrated the feasibility of our method.

6. Related work

The Point Feature Labeling Problem is considered NP Hard because a single change locally in the position of a label may globally affect the positions of many other labels. It could in fact trigger a chain reaction of position changes in many other labels.

Several algorithms have been proposed so far to solve the PFLP problem. Methods have been proposed to solve the problem using Exhaustive Search. Greedy Algorithms, Discrete Gradient Descent, Hirsch's Algorithm and simulated Annealing. All these methods have been empirically studied in [3]. While all these methods have their own advantages and disadvantages there are instances where these algorithms have failed to reduce conflicts between labels.

The Tabu Search Heuristic was proposed by Fred Glover [4], [5]. Similar to other heuristics used to solve the PFLP problem, the Tabu search heuristic does not reduce conflict values among labels to zero under all circumstances. Nevertheless it is by far better than the other heuristics as shown in [6], [7]

7. Conclusion and Future Work

As mentioned previously it is not possible to correct conflicts among labels in all cases using the Tabu Search Algorithm. We see that in Method 5 in both tables 1 and 2. The main reason for the failure of Tabu Search is that there are no places for the move-

Table 3 Analysis - new area brought into frame, focus enabled

Method (considering user's focus area)	No of Labels displayed	No of conflicting labels	Execution time (secs)
Method-1	90	77	0.581
Method-2	29	13	0.311
Method-3	35	0	0.732
Method-4	23	0	0.27
Method-5	90	32	2.073
Method-6	29	0	0.301
Method-7 (our method)	62	0	1.332
Method-8 (our method)	29	0	0.35

Table 4 Analysis - new area brought into frame, focus disabled

Method not considering uesr's focus area	No of Labels displayed	No of conflicting labels	Execution time (secs)
Method-1	477	295	0.631
Method-2	82	30	0.601
Method-3	266	0	4.106
Method-4	64	0	0.391
Method-5	477	147	7.781
Method-6	82	5	0.41
Method-7	383	0	9.974
Method-8	80	0	0.431

ment of the labels around the point where it can be placed without conflicts. Apart from this reason there are also circumstances where conflicts may have possibly been removed but the results obtained are not satisfactory. Such circumstances are explained below:

（1） Even though the conflict value has been brought down to zero, the labels are so close to each other that they are difficult to read and they hide the map features beneath them.

（2） At times the length of the labels are very long, that when the labels move to a different place they seem to have moved a lot away from the original position and hence name a different feature all together.

Though the method of random filtering reduces the amount of conflict to zero there is no guarantee whether all the important labels have been displayed or not. Hence a more credible method will be required for minimizing the conflict values.

The algorithm for the determination of user's focus area can be used when the focus of the user is on one area of the map. The algorithm fails when the focus is on more than one area. Also the focus area cannot be exactly determined when the user deliberately reduces the scale of a particular feature to magnify the surrounding features. Future work will be based on obtaining a solution to the above mentioned problems.

References

[1] Takashi Saruwatari, Yoshihide Hosokawa, Naohisa Takahashi. A Context Aware Map Synthesizer: design and implementation. DBSJ Letters, Vol.2, No.1, pp.67–70 (2003)

[2] Michael Schreyer, Günther R.Raidl. Letting Ants Labeling Point Features. Proceedings of the 2002 IEEE Congress on Evolutionary Computation. pp1564-1569. May 2002

[3] Jon Christensen, Joe Marks, Stuart Schieber. An Empirical Study of Algorithms for Point Feature Label Placement. ACM Transactions on Graphics 1995.

[4] Fred Glover. Tabu Search - part i. ORSA journal on Computing, 1. 190-206, 1989a.

[5] Fred Glover. A user's guide to Tabu Search. Annals of Operations Research 41, 3-28.

[6] Yamamoto, M, G. Camara and L. Lorena. Tabu Search Heuristic for Point Feature Cartographic Label Placement. Geoinformatica,6(1),77-90(2002)

[7] J. Mark Ware, Ian D Wilson, J Andrew Ware, Christopher B Jones A Tabu Search Approach to Automated Map Generalisation. ACM, GIS 2002 Pg 101-106.