International Workshop On
Combinatorial Algorithms
IWOCA

Open Problems in Dynamic Map Labeling

Chee K. Yap
New York University, New York.
yap@cs.nyu.edu

9 May 2012

The problem of map labeling is well-studied in GIS and computational geometry. The standard setting for such problems is the traditional (static) map. We are interested in dynamic maps, the sort we are familiar with on the web. The user can zoom and pan to view different portions of some large map that does not fit in the screen. One idea for solving dynamic labeling is to reduce them directly to static labeling. This is essentially the approach of Petzold [3, 4, 6, 5]. The problem with this approach is the consistency issue as pointed out by [1]. They listed a set of consistency desiderata, formalized the dynamic labeling problem. In particular, they introduce a natural class of dynamic placements called point-invariant placements which has a nice interpretation as a geometric cone (the third dimension is scale space). They also provided a practical solution satisfying the desiderata. This solution is implemented in the G-Vis System [2], a dynamic map for continental USA which is accessible by any browser.

Since this problem is quite new, most issues are open. But here are two specific problems taken from [1]:

1. The solution in [1] assumes each label has a single placement. It should not be hard to generalize this to having a (small) finite set of discrete possibilities per label. More challenging is to accommodate labels with a continuous set of possibilities.

2. In the active range optimization (ARO) problem in [1] we view a dynamic label as a cone and the problem is to truncate these cones so that they are non-overlapping, subject to some optimization criteria. In the 2-D version of the problem the cone has a rectangular base and in the 1-D version the base is a line segment. In a simplified version of the problem a label is never deselected when zooming in. It was shown in [7] that the simplified 1-D problem has a polynomial time solution while in 2-D both the simple and general problems are NP-complete. Approximation algorithms for a number of different variants of the problem were also discussed in [7]. It remains open whether many variants of the 1-D general problem are NP-complete, whether any of the approximation factors can be improved, or whether any of the problem variants admits a polynomial-time approximation scheme.

More Detail Here is the standard (simplified) setting: we are given a map M, viewed as rectangular region of the plane, and containing a set of (map) features. There are three kinds of features:

points, lines, regions. Note that a line feature is really a polygonal line, and a region feature is really a connected polygonal region.

Each of these features has a label (viewed as a floating rectangle) which may be placed on the map satisfying suitable constraints. E.g., A point label must be within a certain radius of the point (but not cover it). A line label must be placed parallel to one on the line segments of the polyline, either above or below the line segment within some distance. A region label must intersect the region.

The computational problem is to (1) select a subset of the labels, and (2) place each selected label in a suitable position satisfying the above constraints, such that no two labels overlap. The optimization criterion is usually to maximize the number of selected labels. Sometimes, we assume the selection subproblem (1) has been solved, and we only want to do the placement subproblem (2).

In the dynamic setting, we assume that only a portion of the map M is visible. That is we can choose a "viewing window or portal P. Intuitively, we can zoom and pan this portal to view different parts of the map at different scales. For simplicity, assume the portal shape is fixed (say, it is a square). A portal is parametrized by three numbers: (s, x, y) where $s > 0$ is the zoom scale, and (x, y) is the portal position. This determines a square $P(s, x, y)$ centered at position (x, y) on M. The sides of $P(s, x, y)$ has length s. We imagine this portion of M is then transformed to a fixed size window W on the computer screen. Note that W has the same shape as the portal, and a label displayed on W has a fixed size (i.e., it does not depend on the scale s). This means that when we zoom in, labels grow in size on the screen, and two non-overlapping labels may begin to overlap.

Petzold et al. [3, 4, 6, 5] used a pre-processing solution that reduces dynamic to static map labeling. The idea is this: given (s, x, y), we first retrieve some superset L of the labels that are potentially visible in the portal $P(s, x, y)$. Then we run a static labeling algorithm on this set L. The problem with this solution is that there is no "consistency for different choices of s, x, y. A set of consistency requirements is specified in [1], who also provided a solution that satisfies them.

References

[1] K. Been, E. Daiches, and C. Yap, Dynamic map labelling, IEEE Transactions on Visualization and Computer Graphics, 12(5):773-780,2006. Proc. 12th Symp. on Information Visualization (InfoVis06), Baltimore, Maryland, October 2006.

[2] G-Vis Dynamic Labeling Demo, 2002. URL http://sage.mc.yu.edu/gvis/.

[3] I. Petzold. Textplazierung in dynamisch erzeugten Karten. Diploma-thesis, Institute for Computer Science, Bonn, 1996.

[4] I. Petzold, Beschrifting von Bildschirmkarten in Echtzeit - Konzept und Struk- tur. Ph.D. thesis, Institute of Cartographt and Geoinformation, University of Bonn,2003.

[5] I. Petzold, G. Gröger, and L. Plümer, Fast screen map labeling - data-structures and algorithms, In Proc. 21th International Cartographic Conferences (ICC'03), pages 288-298, 2003.

[6] I. Petzold, L. Plümer, and M. Heber, Label placement for dynamically generated screen maps. In Proc. 19th International Cartographic Conferences (ICC'99), pages 893-903, 1999. Chee Yap, New York University, USA.

[7] K. Been, M. Nöllenburg and S. Poon, Optimizing active ranges for consistent dynamic map labeling. Computational Geometry, Vol.43, No.3, pp.312-328, 2010. Special Issue on 24th Annual Symposium on Computational Geometry (SoCG'08)