Optimizing Active Ranges for Consistent Dynamic Map Labeling

Ken Been* Martin Nöllenburg† Sheung-Hung Poon‡ Alexander Wolff§

Abstract

Map labeling encounters unique issues in the context of dynamic maps with continuous zooming and panning—an application with increasing practical importance. In *consistent* dynamic map labeling, distracting behavior such as popping and jumping is avoided. In our model a *dynamic label placement* is a continuous function that assigns a 2d-label to each scale. This defines a 3d-solid, with scale as the third dimension. To avoid popping, we truncate each solid to a *single* scale range, called its *active range*. This range corresponds to the interval of scales at which the label is visible. The *active range optimization (ARO)* problem is to select active ranges so that no two truncated solids overlap and the sum of the active ranges is maximized. We show that the ARO problem is NP-complete, even for quite simple solid shapes, and we present constant-factor approximations for different variants of the problem.

1 Introduction

Recent years have seen tremendous improvements in Internet-based, geographic visualization systems that provide continuous zooming and panning (e.g., Google Earth), but relatively little attention has been paid to special issues faced by map labeling in such contexts. In addition to the need for interactive speed, several desiderata for a *consistent dynamic labeling* were identified in [1]: labels do not pop in and out or jump (suddenly change position or size) during panning and zooming, and the labeling is a function of scale and view area—it does not depend on the user's navigation history.

Model. We adapt the following labeling model from [1], with slight changes. In *static labeling*, the key operations are selection and placement—select a subset of the labels that can be placed without overlap. A static placement of a label L is a transformation π^L, composed of translation, rotation, and dilation, that takes L's canonical shape into world coordinates. Once all labels are placed, a viewing transformation takes world coordinates to map coordinates.

In *dynamic labeling* we take scale as an additional dimension. As with [1, 4], we define scale as the inverse of cartographic scale, so that it increases when zooming out. A *dynamic placement* of L is a function that assigns a static placement π_s^L to each scale $s \geq 0$. The translation, rotation and dilation components of the dynamic placement must each be continuous functions of scale. This eliminates jumping and popping during panning, and dependence on navigation history. *Dynamic selection* is similarly a Boolean function of scale. To eliminate popping during zooming we require that each label L_i, $1 \leq i \leq n$, is selected precisely on a single interval of scales, $[a_i, A_i]$, which is called the *active range* of L_i. Thus all consistency desiderata can be satisfied by adhering to this model.

Let $S_{\max}$ be a universal maximum scale for all labels. We define the *available range* of L_i to be an interval of scales, $[s_i, S_i] \subseteq [0, S_{\max}]$, in which label L_i "wants" to be selected. We require $[a_i, A_i] \subseteq [s_i, S_i]$. Since the dynamic placement is continuous with scale, $E_i = \bigcup_{s \in [s_i, S_i]} \pi_s^{L_i}(L_i)$ is a solid defined by sweeping the label shape along a continuous curve that is monotonic in scale, see Fig. 1. We call E_i the *extrusion* of L_i and $T_i = \bigcup_{s \in [a_i, A_i]} \pi_s^{L_i}(L_i)$ its *truncated extrusion.*

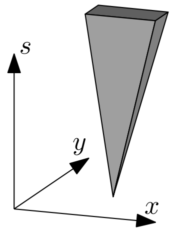


Figure 1: A dynamic label placement is a solid in world coordinates.

The extrusion shapes are determined by the label shape and the translation, rotation and dilation functions that compose the dynamic placement. We restrict our attention to certain classes of extrusions. Our labels are rectangular. For translation, we consider only *invariant point placements*, in which a particular point on the label always maps to the same location in world coordinates, so the label never "slides". Our rotation functions are constant, and yield axis-aligned labels. We consider two classes of dilation functions D^L. If $D^L(s) = bs$ for a constant $b > 0$, then label size is fixed on screen and proportional to scale in world coordinates. The solid is then a label-shaped cone with apex at $s = 0$ as in Fig. 1. With invariant point placements, the cone contains

*Computer Science Department, Yeshiva University, New York, NY, U.S.A. kbeen@yu.edu

†Fakultät für Informatik, Universität Karlsruhe, Germany. noellenburg@iti.uka.de

‡Department of Computer Science, National Tsing Hua University, Hsin-Chu, Taiwan. spoon@cs.nthu.edu.tw

§Faculteit Wiskunde en Informatica, Technische Universiteit Eindhoven, The Netherlands. http://www.win.tue.nl/~awolff

extrusion shape	ARO	dilation	approx.	running time	reference
congruent square cones	simple	bs	1/4	$O((k+n)\log^2 n)$	Theorem 4
congruent square cones		bs	1/8	$O(n\log^3 n)$	Corollary 8
arbitrary square cones		bs	1/24	$O(n\log^3 n)$	Theorem 7
segments of congruent square cones	general	bs	1/4	$O((k+n)\log^2 n)$	Theorem 4
congruent frusta		$bs+c$	$1/(4W)$	$O(n^4)$	Theorem 3

Table 1: Results attained in this paper, where k is the number of pairwise intersections between extrusions and W is the width ratio of top over bottom side.

the vertical line through its apex. The cone might be symmetric to that line (e.g., for labeling a region) or might have a vertical side incident to it (e.g., for labeling a point). Secondly, we consider, in a more general setting, functions of the form $D^L(s) = bs + c$ for constants $b > 0$ and $c \neq 0$. The solid in this case is a portion of a cone with apex at $-c/b$.

Objective. Let $\mathcal{E}$ denote the set of all extrusions, and assume we are given an available range for each. For a set $\mathcal{T}$ of truncated extrusions, define $H(\mathcal{T}) = \sum_{i=1}^{n}(A_i - a_i)$ to be the *total active range height*. This is the same as integrating over all scales the function $f(s)$ that counts the number of labels selected at scale s. The *(general) active range optimization (ARO)* problem is to choose the active ranges so as to maximize H, subject to the constraint that no two truncated extrusions overlap. This is the dynamic analogue of placing the maximum number of labels without overlap in the static case. We call any set of active ranges that correspond to non-overlapping truncated extrusions a *solution*. It is of both theoretical and practical interest to also consider a version of the problem in which $[s_i, S_i] = [0, S_{\max}]$ and $a_i = 0$ for all i. We call this variant of ARO *simple*. In this version all labels want to be selected at all scales, and a label is never deselected when zooming in.

Already the simple ARO problem is NP-complete. Table 1 summarizes the approximation results obtained in this paper. In the full version we also consider 1d-labels, which are segments on the x-axis. The 1d-problem can be seen as a scheduling problem with geometric constraints and is closely related to geometric maximum independent set problems.

Previous Work. Map labeling has been the focus of extensive algorithmic investigation, see the map-labeling bibliography [5]. However, the majority of the research efforts cover static labeling. For dynamic labeling, Petzold et al. [2, 3] use a preprocessing phase to generate a data structure that is searched during interaction to produce a labeling for the current scale and view area. Poon and Shin [4] build a hierarchy of precomputed solutions, and interpolation between these produces a solution for any scale. Neither of these approaches satisfies the consistency desiderata. In addition to introducing consistency for dynamic map labeling, Been et al. [1] show that simple ARO is NP-complete for star-shaped labels, and implement a simple heuristic solution in a working system.

2 Complexity

Already the simple ARO problem for congruent square cones as extrusion shape is NP-complete. The proof is by reduction from Planar3SAT using 3d-gadgets. We omit it here due to space constraints.

Theorem 1 *Simple ARO with proportional dilation is NP-complete, i.e., given a real $K > 0$ and a set $\{E_1, \ldots, E_n\}$ of congruent square cones, it is NP-complete to decide whether there is a set of truncated extrusions $\mathcal{T} = \{T_1, \ldots, T_n\}$ with $T_1 \subseteq E_1, \ldots, T_n \subseteq E_n$ and $H(\mathcal{T}) \geq K$.*

3 Approximation algorithms

In this section we give two algorithms that yield constant-factor approximations for a number of different variants of the ARO problem. The first algorithm in Sect. 3.1 is based on sweeping the extrusions from top to bottom and the second one in Sect. 3.2 is a level-based greedy algorithm. Due to space constraints we omit the proofs of the running times.

3.1 Top-to-bottom fill-down sweep

Algorithm 1 below is based on the idea to sweep down over the extrusions in $\mathcal{E}$, and if $E_i \in \mathcal{E}$ is selected at some height s, we "fill" E_i from s down to its bottom—i.e., we set $[a_i, A_i] = [s_i, s]$. Thus we have $a_i = s_i$ for every E_i that contributes to the objective function H at all.

Say that E_i is *available* if its available range includes the current sweep scale s, and *active* if its active range has already been set and covers s. We are interested in event points at which the conflict graph over the available extrusions changes. This happens at each S_i and s_i, and with some extrusion shapes it also happens at additional heights. If E_i and E_j are both available at s and at $s' > s$, and they intersect at s' but not at s, then let s_{ij} refer to the lowest scale at which they intersect. Let k be the number of s_{ij} events over $\mathcal{E}$. We make use of a subroutine, "try to pick" E_i, which means, "if E_i does not intersect the interior of any extrusion already chosen to be active at the current sweep height s, then make E_i active and set $[a_i, A_i] = [s_i, s]$".

Algorithm 1 *Top-to-bottom sweep algorithm.*
Sweep a plane from top to bottom. At each event point of type S_i, s_i, or s_{ij}, try to pick each available but inactive extrusion E_j, in non-increasing order of S_j.

The following lemma will help proving approximation factors. Let $\mathcal{A} = \{(a_i, A_i)\}$ be the solution computed by Algorithm 1. Say that E_j *blocks* E_i at scale s under a given solution if E_i and E_j overlap (i.e., their interiors intersect) at s and $s \in [a_j, A_j]$. Note that this implies that $s \notin [a_i, A_i]$. Say that two extrusions are *independent at* s if their restrictions to the horizontal plane at height s are non-overlapping.

Lemma 2 *If, for any $E \in \mathcal{E}$ and $s \geq 0$, E can block no more than c pairwise independent extrusions at s, then $\mathcal{A}$ is a $(1/c)$-approximation for the maximum total active range height of $\mathcal{E}$.*

Proof. Suppose that $E \in \mathcal{E}$ is inactive at scale s under $\mathcal{A}$. Then E must be blocked at the nearest event point above (or at) s, since otherwise it would be picked by Algorithm 1. Since the extrusion conflict graph only changes at event points, E is blocked at s. Thus, in $\mathcal{A}$, if E is inactive at any scale s then E is blocked at s.

If at any scale no extrusion can block more than c pairwise independent extrusions, and in $\mathcal{A}$ every inactive extrusion is blocked, then at any scale the number of active extrusions in an optimal solution can be no more than c times the number in $\mathcal{A}$. Integrating over all scales proves the lemma. □

Congruent frusta. The top-to-bottom nature of Algorithm 1 ensures that if a frustum E_j blocks another frustum E_i at scale s then E_i intersects a side face of E_j. The number of independent frusta that can intersect a single face depends on W, the ratio of the side length of the top face of each frustum to that of the bottom face.

Theorem 3 *Algorithm 1 computes a $1/(4W)$-approximation for the maximum total active range height of a set of n congruent frusta in $O(n^4)$ time.*

Frustal segments of congruent square cones. For congruent underlying square cones the size of all squares is the same at each scale. Thus any extrusion blocked by an extrusion E at scale s must intersect one of the four corner edges of E at s, so at most four such extrusions can be independent. The approximation factor in Theorem 4 follows from Lemma 2.

Theorem 4 *Given a set of n frustal segments of axis-aligned unit square cones, Algorithm 1 computes a $(1/4)$-approximation for the maximum total active range height in $O((n + k) \log^2 n)$ time.*

Note that simple ARO with congruent square cones is a special case of the above where each $[s_i, S_i] = [0, S_{\max}]$, so that Theorem 4 still holds in this case.

3.2 Level-based small-to-large greedy algorithm

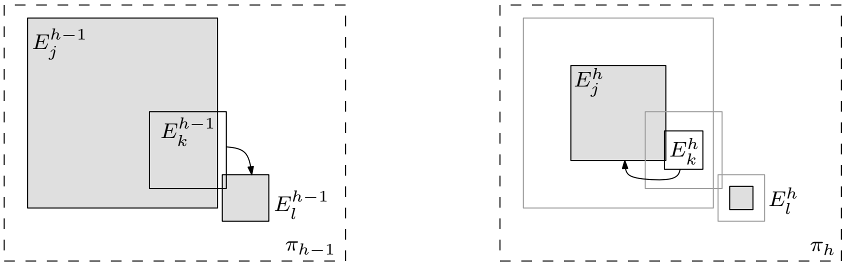
In this section we give an algorithm for simple ARO with square cones. It computes a 1/8-approximation when the cones are congruent, and a (1/24)-approximation otherwise. The algorithm intersects the given cones with $O(\log n)$ horizontal planes, starting at $S_{\max}$ and proceeding downward.

Algorithm 2 *Level-based algorithm for 3d-cones*
Initially no extrusion is active. In phase i, $i = 0, \ldots, \lceil \log n \rceil$, let π_i be the horizontal plane at scale $s = S_{\max}/2^i$. Let E_j^i be the intersection of extrusion E_j with π_i and call E_j^i *active* if E_j is already active. As long as there is an inactive object E_j^i that does not intersect any active object, choose the smallest such object $E_{j^\star}^i$ and make $E_{j^\star}$ (and $E_{j^\star}^i$) active by setting $A_{j^\star} = s$.

We first consider arbitrary square cones that are symmetric to the vertical axes passing through their apexes. When the algorithm terminates, all squares at level i that are not active must intersect an active square—they are *blocked*. We associate each blocked square E_j^i to one of the active squares in the following way: (i) If E_j^i was not blocked at the beginning of phase i but became blocked by a newly activated square E_k^i, then associate E_j^i to E_k^i. (ii) If E_j^i was blocked in the beginning of phase i then associate E_j^i to any of its blocking squares that were active at the beginning of phase i. Next, we show that the squares associated to an active square cannot be arbitrarily small.

Lemma 5 *Let E_j^i be an active square at level i with side length ℓ_j^i. Then any square associated to E_j^i has side length at least $\ell_j^i/3$ and intersects the boundary of E_j^i.*

Proof. Let E_k^i be associated to E_j^i with $\ell_k^i < \ell_j^i$. By the greedy choice of the algorithm, all squares associated to a newly active square are larger than it. This implies that E_j must have been activated at a higher level, and that E_k must have been reassigned to E_j at some level $h \leq i$. Thus, at level $h-1$ square E_k^{h-1}



(a) Squares at level $h-1$. (b) Squares at level h.

Figure 2: Intersection behavior of E_j, E_k, E_l at two consecutive levels.

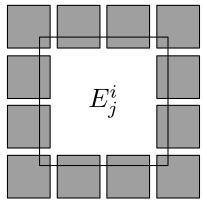


Figure 3: At most 12 independent squares intersect E_j^i.

was associated to another square E_l^{h-1}. Note that for this reassignment to take place at level h, E_j^{h-1} must have been active. Thus we know that E_j^{h-1} and E_l^{h-1} do not intersect, but they both intersect E_k^{h-1}; see Fig. 2a. At level h the reassignment takes place because E_k^h no longer intersects E_l^h but still intersects E_j^h; see Fig. 2b. Now suppose $\ell_k^h < \ell_j^h/3$. Then by going from level h to $h-1$ the side lengths of the squares are doubled and it is easy to verify that E_k^{h-1} would be contained in E_j^{h-1}, a contradiction to the fact that $E_k^{h-1} \cap E_l^{h-1} \neq \emptyset$. As $\ell_k^h \geq \ell_j^h/3$ this also holds for level i, and since E_k^{h-1} intersects the boundary of E_j^{h-1} this is also still true for level i. □

Let $\pi_{\lceil \log n \rceil + 1}$ be the plane $s = 0$, and denote the active segments of the extrusions in the optimal solution $\mathcal{S}$ and our algorithm's solution $\mathcal{A}$ between planes π_{i-1} and π_i by $\mathcal{S}_i$ and $\mathcal{A}_i$, respectively. We charge the active range height $H(\mathcal{S}_i)$ to that of $H(\mathcal{A}_{i+1})$.

Lemma 6 *For $i \in 1, \ldots, \lceil \log n \rceil - 1$ it holds that $H(\mathcal{A}_{i+1}) \geq 1/24\, H(\mathcal{S}_i)$.*

Proof. Let square E_j^i be active in $\mathcal{A}$ and consider the set $D(E_j^i)$ of squares in π_i associated to it. The squares in $D(E_j^i)$ that correspond to active extrusions in $\mathcal{S}_i$ cannot intersect each other.

By Lemma 5, all squares in $D(E_j^i)$ have side length at least $\ell_j^i/3$ and intersect the boundary of E_j^i. Thus, at most 12 of those squares can be independent in π_i and hence active in $\mathcal{S}_i$ like in Fig. 3. Now the height between levels i and $i-1$ is twice the height between levels $i+1$ and i. Hence the active height of E_j in $\mathcal{A}_{i+1}$ is at least $1/24$ times the sum of heights of active extrusions in $\mathcal{S}_i$ whose squares at level i are associated to E_j^i. It follows that $H(\mathcal{A}_{i+1}) \geq 1/24\, H(\mathcal{S}_i)$. □

Theorem 7 *Algorithm 2 computes a $(1/24)$-approximation to the maximum total active range height of a set of arbitrary square cones in $O(n \log^3 n)$ time.*

Proof. From Lemma 6, it remains to compare $H(\mathcal{S}_{\lceil \log n \rceil}) + H(\mathcal{S}_{\lceil \log n \rceil + 1})$ to $H(\mathcal{A}_{\lceil \log n \rceil + 1}) + H(\mathcal{A}_1)$. The height of $\pi_{\lceil \log n \rceil - 1}$ is at most $2S_{\max}/n$ and obviously there are at most n active cone segments in $\mathcal{S}$ below $\pi_{\lceil \log n \rceil - 1}$, so their total active range height is at most $2S_{\max}$. On the other hand, there is at least one active cone segment in $\mathcal{A}_1$ of height $S_{\max}/2$. Thus the approximation factor is indeed $1/24$. □

With congruent square cones, all squares at each level are the same size, so at most four rather than 12 independent squares can intersect a given square. A similar argument gives the following corollary.

Corollary 8 *Algorithm 2 computes a $(1/8)$-approximation to the maximum total active range height of a set of congruent square cones in $O(n \log^3 n)$ time.*

4 Conclusions

ARO is an exciting new problem inspired by interactive web-based mapping applications and we have given approximation algorithms for some variants. It remains open whether any of the problems admits a PTAS. Also, mapping applications in practice often require more complex extrusion shapes.

References

[1] K. Been, E. Daiches, and C. Yap. Dynamic map labeling. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):773–780, 2006.

[2] I. Petzold, G. Gröger, and L. Plümer. Fast screen map labeling—data-structures and algorithms. In *Proc. 23rd Internat. Cartographic Conf. (ICC'03)*, pages 288–298, Durban, South Africa, 2003.

[3] I. Petzold, L. Plümer, and M. Heber. Label placement for dynamically generated screen maps. In *Proc. 19th Internat. Cartographic Conf. (ICC'99)*, pages 893–903, 1999.

[4] S.-H. Poon and C.-S. Shin. Adaptive zooming in point set labeling. In M. Liśkiewicz and R. Reischuk, editors, *Proc. 15th Internat. Sympos. Fundam. Comput. Theory (FCT'05)*, volume 3623 of *Lecture Notes Comput. Sci.*, pages 233–244. Springer-Verlag, 2005.

[5] A. Wolff and T. Strijk. The Map-Labeling Bibliography. `http://i11www.ira.uka.de/map-labeling/bibliography`, 1996.