Dynamic Map Labeling

Ken Been, Eli Daiches & Chee Yap

Dynamic maps allow panning and zooming.

Problem: Selecting and placing labels on a dynamic map.

vs. Static Label Placement

- 2D => 3D
- HCI
- performance requirements

Basic cartographic rules for map labels:

- No overlap among labels and features
- Each label identifies a unique feature
- Each label optimum among alternatives

Label Placement

- Static Case: Transformation π of a box in label-coordinates into world coordinates

- Dynamic Case: A parameterized family of transformations $\Pi = \{\pi_s : s \in [\lambda_0, \lambda_h]\}$

Zoom levels
lowest to highest

Label Placement

Given a set of labels S_0

Two-step Process

1. Select a subset $S \subseteq S_0$
2. Place the labels S on the map

Interactive speeds non-trivial!

Label Placement

Given a set of labels S_0

0. label-filtering $S' \subseteq S_0$

Two-step Process

1. Select a subset $S \subseteq S'$
2. Place the labels S on the map

Interactive speeds non-trivial!

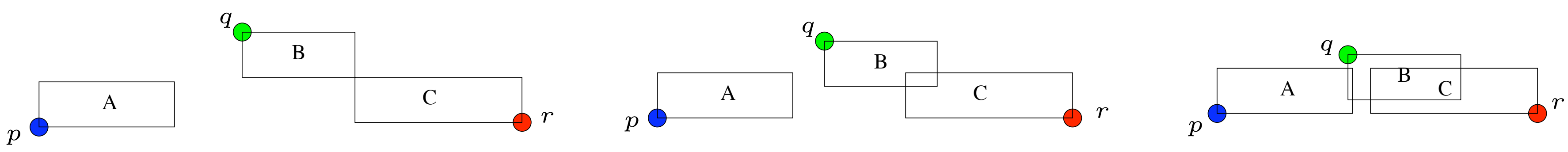
Previous Work

- 1980s - static placements
- 1990s - interactive
  - show label when mouse over
  - draw lines from labels to features around mouse
- Petzold, et al.(96, 03)
  - Preprocessing phase + Interaction phase
  - Label selection based on priorities
  - realtime achievable
- Zhang & Harrie(04)
  - selection and placement for points and lines.
  - not realtime

Label Consistency

Label Size Invariance Property: Each label on screen has a fixed size that is invariant under zooming

(D1) Labels should not vanish when zooming in, and not appear when zooming out



(i) Zoom level s_1 (ii) Zoom level S_2 (iii) Zoom level s_3

label priority A>B>C, zoom levels $s_1<s_2<s_3$

Label Consistency

Label Size Invariance Property: Each label on screen has a fixed size that is invariant under zooming

(D1) Labels should not vanish when zooming in, and not appear when zooming out.

(D2) The distance a map feature f and position of its label L should vary monotonically.

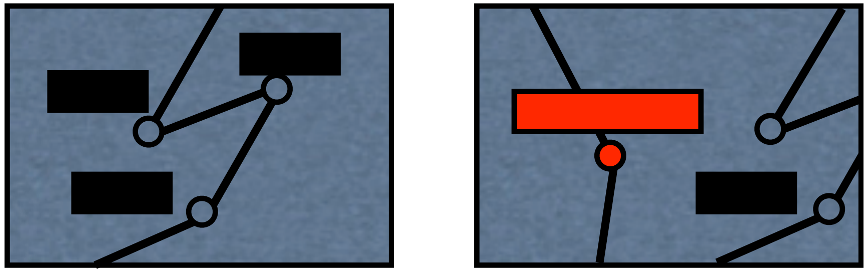
Label Consistency

Label Size Invariance Property: Each label on screen has a fixed size that is invariant under zooming

(D1) Labels should not vanish when zooming in, and not appear when zooming out.

(D2) The distance a map feature f and position of its label L should vary monotonically.

(D3) Labels must not vanish or appear during panning except through sliding in/out of view



Label Consistency

Label Size Invariance Property: Each label on screen has a fixed size that is invariant under zooming

(D1) Labels should not vanish when zooming in, and not appear when zooming out.

(D2) The distance a map feature f and position of its label L should vary monotonically.

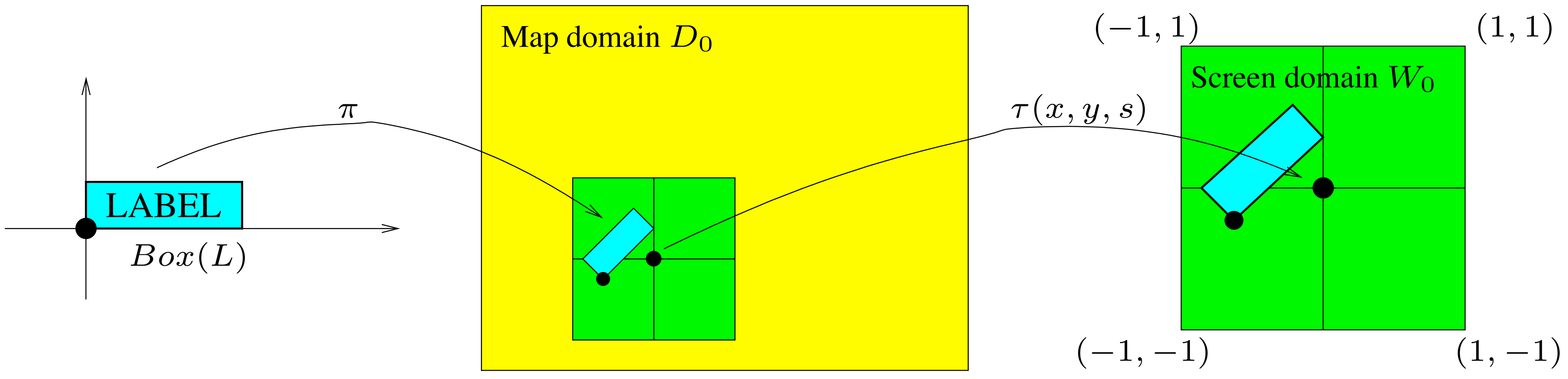
(D3) Labels must not vanish or appear during panning except through sliding in/out of view

(D4) Display of any label is a function of state (x,y,s) so not dependent on how the view was obtained

Dynamic Map Model

- View
  - shape = w x h
  - position = center point (x,y)
  - zoom level = map scale ratio s:1
- Map features - point, line, area
- Label L
  - map feature $\phi(L)$
  - box(L)
  - live range $R(L) \subseteq [0, \infty] \Rightarrow R(L)=[\lambda_i,\lambda_j]$, $i<j$

Label - World - Screen Coordinate Systems



Dynamic Labeling Model

Allowable transformation:

composition of T(e,f), R(φ), D(s)

Static Placement:

allowable transformation from label coords to world coords

Valid Placement π

If L a point label

(P1) π is a translation composed with dilation

(P2) Interior of π(box(L)) does not intersect ϕ(L)

(P3) Distance b/n π(box(L)) and ϕ(L) <= sd_0.

Valid Placement π

If L a area label

(A1) π is a translation composed with dilation

(A2) Interior of $\pi(\mathrm{box}(L))$ must intersect $\phi(L)$

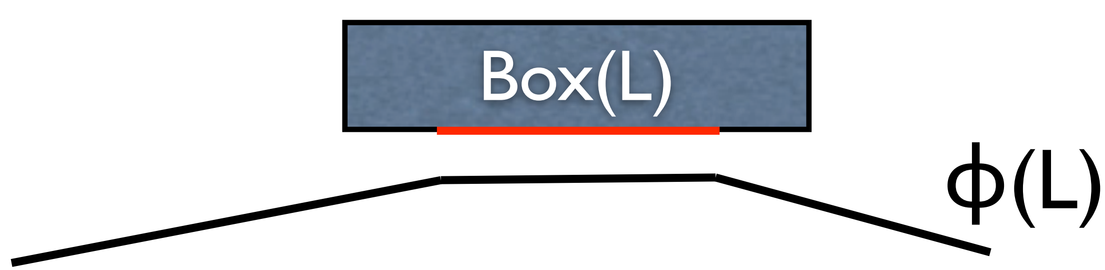
Valid Placement π

If L a line label

(L1) π is allowable

(L2) Interior of π(box(L)) does not intersect ϕ(L)

(L3) Distance b/n π(box(L)) and ϕ(L) <= sd_l. This distance is also achieved at all points on an interval with size at least 1/2 the length of Box(L)



For simultaneous placements $\{\pi^L : L \in S\}$, label-label overlap and label-point feature overlap not allowed

Dynamic Placement : Family $\Pi = \{\pi_s : s \in [\lambda_0, \lambda_h]\}$ of static placements, s.t.

1) $\det(\pi_s) = s^2$

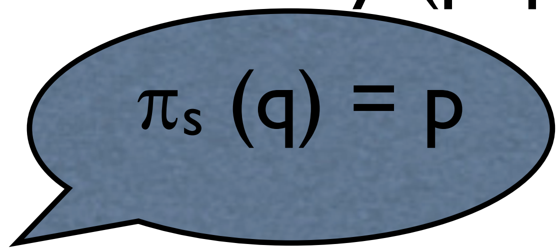
2) π_s varies continuously with s.

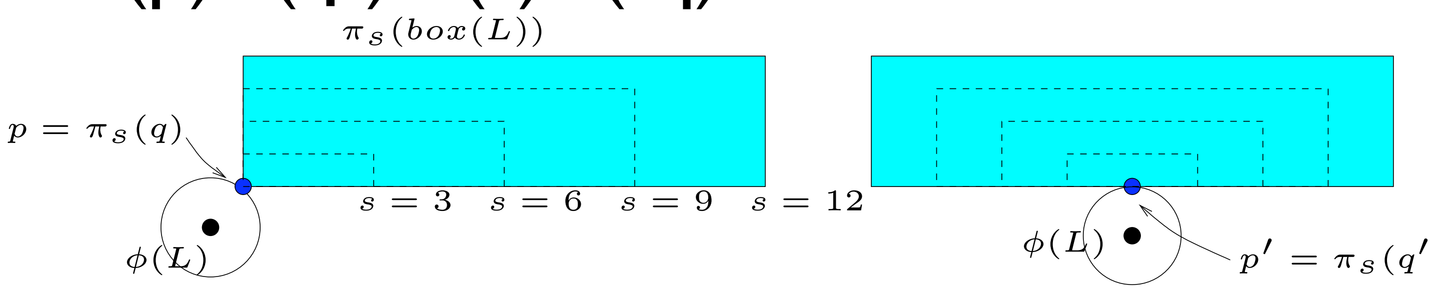
Invariant Point Placement

Dynamic placement represented by (p,q, φ)

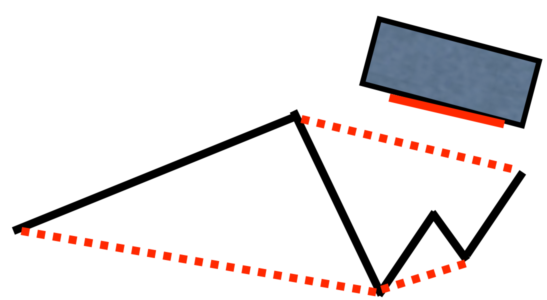
For every zoom level s:

$\pi_s = T(p)R(\varphi)D(s)T(-q)$



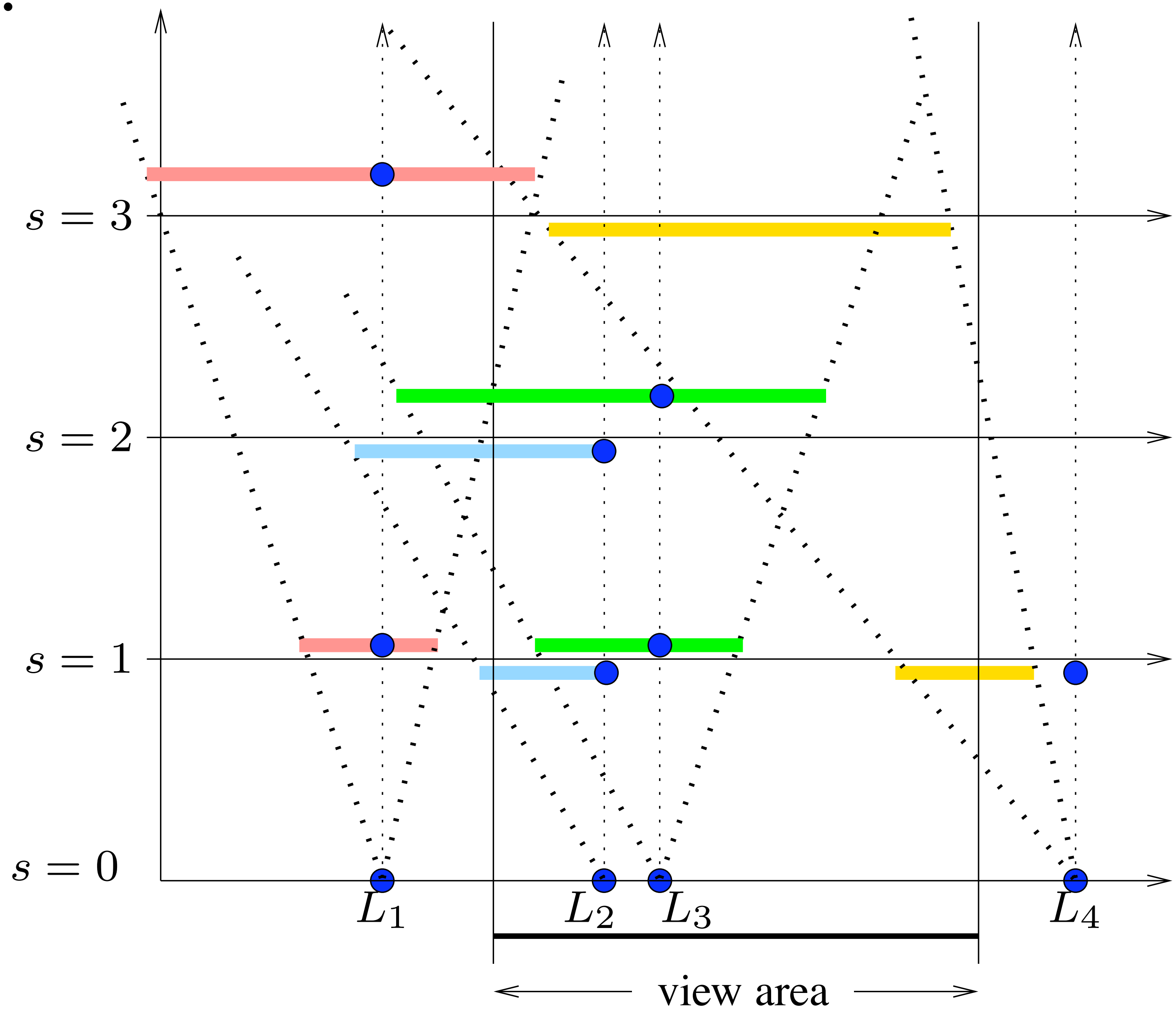


Boundary placement



Geometric View - cones in 3D

in 1D:



New Approach

- S_0 : set of all labels
- Pre-Processing - allows no conflict computation during interactive phase

(1) Compute a set of dynamic placements

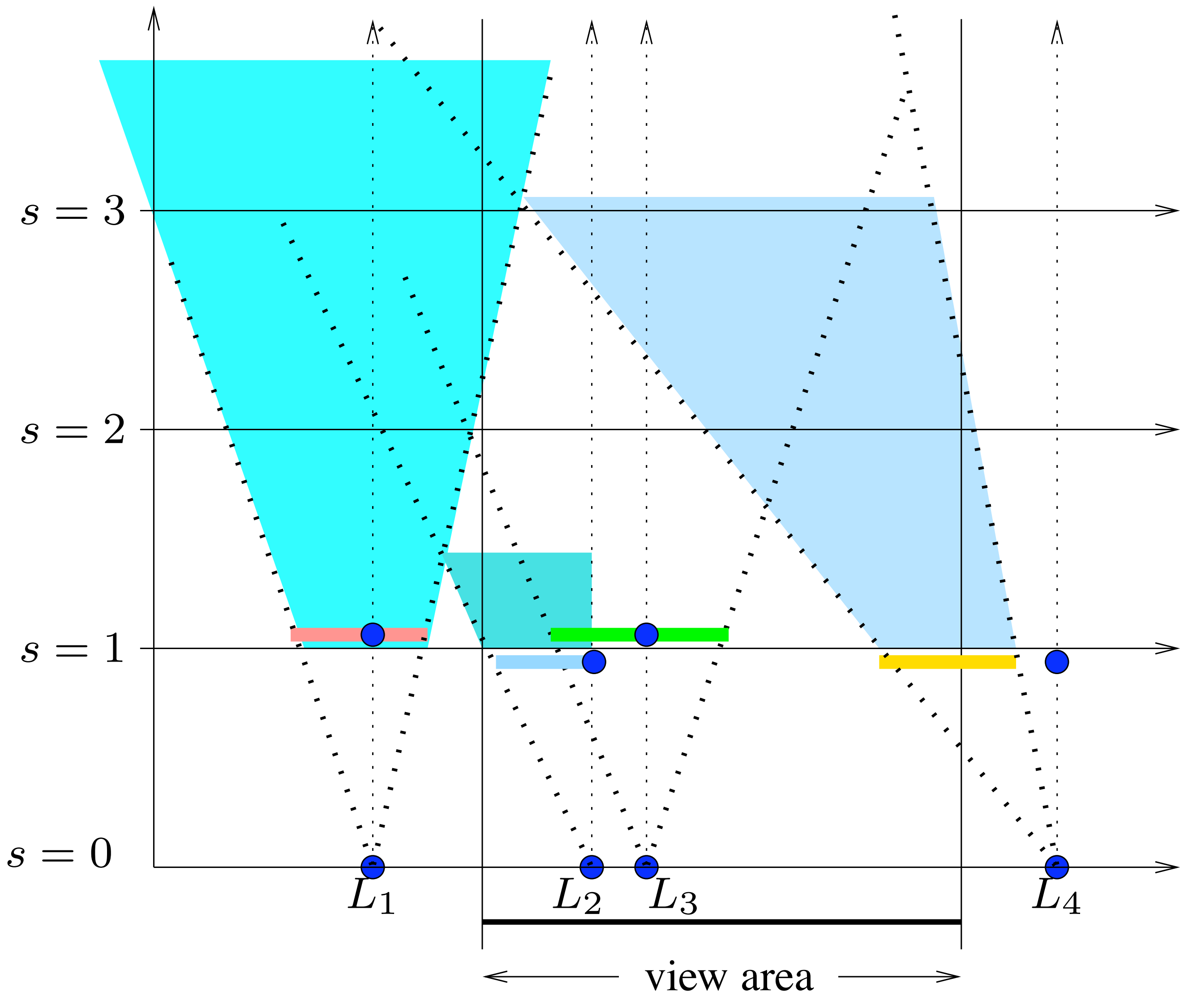
$$\Pi^* = \{\Pi^L : L \in S_0\}$$

(2) Compute a subinterval $A(L) \subseteq R(L), \ \forall L \in S_0$

Active Range

L is active if the current zoom level $s \in A(L)$

- Active Ranges reduces cones to truncated cones
- in 1D : trapezoids.
  Non-intersecting trapz = non overlapping labels



New Approach

- Interactive Phase
  For each L visible in the current view
  If Active then Place

$S_w \subseteq S_0$ be set of labels that intersect W(x,y,s)

$S_a \subseteq S_0$ be set of labels active at level s

$S_v = S_w \cap S_a$, visible labels

a) Filter S_w from S_0
b) Select S_v from S_w
} Slow

Solution: Assume every active range has form

$$A(L) = [0,h]$$

Goal: Produce S_w as a list sorted by h s.t. traversal of list until label L with $h_L < s = S_v$

Method:

- Divide D_0 into a grid of buckets
- Assign each label to the bucket that contains the invariant point.
- In each bucket, sort labels by h_L
- Store buckets in a hash table

Complexity of Optimal Active Range Selection (Pre-Processing)

Optimization Problem: $$|A| := \sum_{i=1}^{n} |I_i| \rightarrow max$$

I_i: intervals in s (cone truncation) w/ no overlap

No solution known even for 1D

simple active range optimization (I_i = [0,h] (full cones))

= height maximization problem on graphs

Height Maximization on Graphs

G(V,E) undirected with functions
$H: V \to R_{\geq 0}$ and $W: V \to R_{\geq 0}$

A height function (H) is compatible if:
(1) $h(v) <= H(v)$
(2) for each edge e(u,v), $h(u) > W(e) \Rightarrow h(v) \leq W(e)$

Problem: Compute a height function that maximizes the sum $|A| := \sum_{v \in V} A(v)$

Reduction:

V = cones, $(u,v) \in E$ iff $u \cap v \neq \emptyset$
W(u,v) = height at which u,v intersects
H(v) = height of cone v

Lemma:

Given an instance of height maximization, and a constant k, it is NP-hard to decide if there exiss a compatible height function A s.t. |A|>=k

Then?

Lemma:

Given an instance of height maximization, and a constant k, it is NP-hard to decide if there exists a compatible height function A s.t. $|A| \geq k$

Then?

Assign Priorities!

Prioritized Labels

- Unique $C_i \neq C_j \Rightarrow P(C_i) \neq P(C_j)$
- $A(C_i) = h_i$ means active range for $C_i = [0,h_i]$
- $C_i \backslash h_i$ is truncated cone $C_i \backslash [0,h_i]$
- A maximal if
  1. C_i/h_i blocked by $C_j/h_j \Rightarrow P(C_j)>P(C_i)$
  2. For all C_i, either C_i blocked or $A(C_i) = R(L_i)$

Lemma: For any set of invariant point placements with priority P, maximal non-overlapping active range function A is unique

Using conflict graphs - can be computed in $O(n+m)$

Conclusions

- Theoretical
  - ✓ Formulated dynamic labeling problem
  - ✓ Desiderata (D1-D4) for consistency
  - ✓ Invariant point labeling
- Algorithmic
  - ✓ Active range computation efficient in interaction phase and satisfies consistency
- Practical
  - ✓ Implementation

  full scale, web demo
  (line features)