We will present a framework for geometric problems

Definitions:

- *Objects*: members of a universe $\mathcal{O}$.
    - $\mathcal{O}$ is usually infinite.
    - The input to a problem is a finite $\mathcal{S} \subset \mathcal{O}$.
    - Objects are point, lines, segments, planes, etc.
- *Region*: members of a set $\mathcal{F}$ of regions.
    - Objects can *determine* a regions
        - Regions are determined by up to b objects, where b is a constant related to the problem.
    - Objects can *conflict* with a region.
        - The set of objects that conflict with a region is the *domain of influence* of the region. Usually infinite.
- *Random r-sample:* a subset $\mathcal{R} \subset \mathcal{S}$ with r elements, randomly chosen from $\mathcal{S}$ with equal probability of $1/\binom{n}{r}$.

Let $\mathcal{S}$ be a set of objects and $F \in \mathcal{F}$ be a region.

- F is *defined over* $\mathcal{S}$ if a set of object that determine it is contained in $\mathcal{S}$.
- F is *without a conflict over* $\mathcal{S}$ if its domain of influence and $\mathcal{S}$ are disjoint.
- F *have* j *conflicts over* $\mathcal{S}$ if its domain of influence has k elements from $\mathcal{S}$.

A geometric problem is defined in terms of objects, regions and conflict in such a way that solving it is equivalent to finding all regions without conflicts.

Example: convex hull in $\mathbb{E}^d$

- $\mathcal{O}$ is the set of all points in $\mathbb{E}^d$.
- $\mathcal{S}$ is a set of n points.
- $\mathcal{F}$ is the set of all half-spaces.
- A half-space is determined by d points.
- A half-space is in conflict with a point if it contains the point.
- Every d points determine *two* half-spaces

Notation:

- $\mathcal{F}(\mathcal{S})$*:* the set of regions defined over $\mathcal{S}$.
- $\mathcal{F}_j(\mathcal{S})$*:* the set of regions with j conflicts defined over $\mathcal{S}$.
- $\mathcal{F}_j^i(\mathcal{S})$*:* the set of regions defined by i objects of $\mathcal{S}$with j conflicts defined over $\mathcal{S}$.
- $f_j^i(\mathcal{S}) = \left|\mathcal{F}_j^i(\mathcal{S})\right|$
- Let $g(\mathcal{R})$ be a function of a sample $\mathcal{R}$. Define $g(r, \mathcal{S})$ to be the expected value of $g(\mathcal{R})$ for a random r-sample of $\mathcal{S}$.
    - Example: $f_j(r, \mathcal{S})$ is the expected number of regions defined and with j conflicts over a random r-samples of $\mathcal{S}$.

The sampling theorem

The sampling theorem gives an upper bounds on the number of regions defined and with at most k conflicts over a set $\mathcal{S}$ of n elements ($f_k(\mathcal{S})$).

Lemma 4.2.1: let $\mathcal{S}$ be a set of n objects and $F \in \mathcal{F}_j^i(\mathcal{S})$. If $\mathcal{R}$ is a r-sample of $\mathcal{S}$, the probability $p_{j,k}^i(r)$ that $F \in \mathcal{F}_k(\mathcal{R})$

$$p_{j,k}^{i}(r)=\frac{\binom{j}{k}\binom{n-i-j}{r-i-k}}{\binom{n}{r}}$$

קודם בוחרים k מתוך ה-j קונפליקטים. ה-i שקובעים צריכים להישאר ואז יש את כל השאר.

<u>Lemma 4.2.2:</u> let $\mathcal{S}$ be a set of n objects and $\mathcal{R}$ a random sample of $\mathcal{S}$. Then $f_k^i(r,\mathcal{S})$ is

$$f_k^i(r,\mathcal{S})=\sum_{F\in\mathcal{F}^i(\mathcal{S})}P\left(F\in\mathcal{F}_k^i(\mathcal{R})\right)=\sum_{j=0}^{n-i}\sum_{F\in\mathcal{F}_j^i(\mathcal{S})}p_{j,k}^i(r)=\sum_{j=0}^{n-i}\left|\mathcal{F}_j^i(\mathcal{S})\right|p_{j,k}^i(r)=\sum_{j=0}^{n-i}\left|\mathcal{F}_j^i(\mathcal{S})\right|\frac{\binom{j}{k}\binom{n-i-j}{r-i-k}}{\binom{n}{r}}$$

תזכורת: $f_k^i(\mathcal{S})=|\mathcal{F}_k^i(\mathcal{S})|$.

<u>Theorem 4.2.3 (Sampling Theorem)</u>: let $\mathcal{S}$ be a set of n objects and and k an integer such that $2\le k\le\frac{n}{b+1}$. Then

$$|\mathcal{F}_{\le k}(\mathcal{S})|\le 4(b+1)^b k^b f_0(\lfloor n/k\rfloor,\mathcal{S})$$

נתחיל מ

$$\left|\mathcal{F}_{\le k}^i(\mathcal{S})\right|\le 4(b+1)^i k^i f_0^i(\lfloor n/k\rfloor,\mathcal{S})$$

מהלמה

$$f_0^i(r,\mathcal{S})=\sum_{j=0}^{n-i}\left|\mathcal{F}_j^i(\mathcal{S})\right|\frac{\binom{n-i-j}{r-i}}{\binom{n}{r}}\ge\sum_{j=0}^{k}\left|\mathcal{F}_j^i(\mathcal{S})\right|\frac{\binom{n-i-k}{r-i}}{\binom{n}{r}}=\left|\mathcal{F}_{\le k}^i(\mathcal{S})\right|\frac{\binom{n-i-k}{r-i}}{\binom{n}{r}}$$

אפשר להראות ש-

$$\frac{\binom{n-i-k}{r-i}}{\binom{n}{r}}\ge\frac{1}{4(b+1)^i k^i}$$

ובזה סיימנו

<u>Corollary 4.2.4</u>

$$f_1(r,\mathcal{S}),f_2(r,\mathcal{S})\le\beta f_0(\lfloor r/2\rfloor,\mathcal{S})$$

For r such that $n\ge r\ge 2(b+1)$ and $\beta=4(b+1)^b2^b$.

הוכחה:

$$|\mathcal{F}_1(R)|\le 4(b+1)^b2^b f_0(\lfloor n/2\rfloor,\mathcal{R})$$

ולקחת תוחלת של שני הצדדים.

<u>The moment theorem</u>

The moment of order k of $\mathcal{R}$ with respect to $\mathcal{S}$ is

$$m_k(\mathcal{R},\mathcal{S}) = \sum_{F\in\mathcal{F}_0(\mathcal{R})} \binom{|\mathcal{S}(F)|}{k}$$

Examples:

$$m_0(\mathcal{R},\mathcal{S}) = |\mathcal{F}_0(\mathcal{R})| = \text{number of regions without conflicts}$$

$$m_1(\mathcal{R},\mathcal{S}) = \sum_{F\in\mathcal{F}_0(\mathcal{R})} |\mathcal{S}(F)| = \text{total number of conflicts}$$

Lemma 4.2.5

$$m_k(r,\mathcal{S}) = \sum_{i=1}^{b}\sum_{j=0}^{n-i} \left|\mathcal{F}_j^i(\mathcal{S})\right| \binom{j}{k} p_j^i(r)$$

Theorem 4.2.6 (Moment theorem)

$$m_k(r,\mathcal{S}) \le f_k(r,\mathcal{S}) \frac{(n-r+k)!\,(r-b-k)!}{(n-r)!\,(r-b)!}$$

הוכחה:

$$m_k(r,\mathcal{S}) = \sum_{i=1}^{b}\sum_{j=0}^{n-i} \left|\mathcal{F}_j^i(\mathcal{S})\right| \binom{j}{k} \frac{\binom{n-i-j}{r-i}}{\binom{n}{r}} \le \frac{(n-r+k)!\,(r-b-k)!}{(n-r)!\,(r-b)!} \sum_{i=1}^{b}\sum_{j=0}^{n-i} \left|\mathcal{F}_j^i(\mathcal{S})\right| \binom{j}{k} \frac{\binom{n-i-j}{r-i-k}}{\binom{n}{r}}$$

תזכורת

$$\sum_{j=0}^{n-i} \left|\mathcal{F}_j^i(\mathcal{S})\right| \binom{j}{k} \frac{\binom{n-i-j}{r-i-k}}{\binom{n}{r}} = f_k^i(r,\mathcal{S})$$

ולכן הכל נובע.

Corollary 4.2.7: there exists a real constant γ and an integer r_0 such that for each $r_0 \le r \le n$,

$$m_1(r,\mathcal{S}), m_2(r,\mathcal{S}) \le \gamma \frac{(n-r)^{1,2}}{r^{1,2}} f_0(\lfloor n/k \rfloor, \mathcal{S})$$

פשוט מציבים

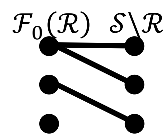
Randomized algorithms – the randomized incremental method

Process the objects one by one in a random order

$$\mathcal{F}_0(\mathcal{R}) \to \mathcal{F}_0(\mathcal{R} \cup \{O\})$$

- If $F \in \mathcal{F}_0(\mathcal{R})$ but $F \notin \mathcal{F}_0(\mathcal{R} \cup \{O\})$ then F was *killed* by O.
- If $F \notin \mathcal{F}_0(\mathcal{R})$ but $F \in \mathcal{F}_0(\mathcal{R} \cup \{O\})$ then F was *created* by O.

The conflict graph



Update Condition 5.2.1

1. Updating $\mathcal{F}_0(\mathcal{R})$ can be carried in time proportional to the number of regions killed or created.
2. Updating the graph can be carried in time proportional to the number of conflicts added or removed.

Lemma 5.2.2: let $\mathcal{S}$ be a set of n objects and $F \in \mathcal{F}_j^i(\mathcal{S})$

1. The probability $p_j^{i\prime}$ that F is created by a randomized incremental algorithm processing $\mathcal{S}$ is
$$p_j^{i\prime} = \frac{i!\,j!}{(i+j)!}$$
2. The probability $p_j^{i\prime}(r)$ that F is created at step r is
$$p_j^{i\prime}(r) = \frac{i}{r} p_j^i(r)$$

Theorem 5.2.3 (Conflict graph)

1. The expected total number of regions created is
$$O\left(\sum_{r=1}^{n} \frac{f_0(r,\mathcal{S})}{r}\right) \text{ if } f_0(r,S) = O(n), \big(O(n)\big)$$
$$v(\mathcal{S}) = \sum_{i,j} \left|\mathcal{F}_j^i(\mathcal{S})\right| p_j'^i = \sum_{i,j} \left|\mathcal{F}_j^i(\mathcal{S})\right| \sum_{r=1}^{n} p_j'^i(r) = \sum_{i,j} \sum_{r=1}^{n} \left|\mathcal{F}_j^i(\mathcal{S})\right| \frac{i}{r} p_j^i(r) = O\left(\sum_{r=1}^{n} \frac{f_0(r,\mathcal{S})}{r}\right)$$
2. The expected total number of conflicts arc added
$$O\left(n \sum_{r=1}^{n} \frac{f_0(r,\mathcal{S})}{r^2}\right) O(nlogn)$$
$$e(\mathcal{S}) =$$

קשור למשפט המומנט...

3. If the algorithm satisfies the update condition, then the complexity (time and space) is on average
$$O\left(n \sum_{r=1}^{n} \frac{f_0(r,\mathcal{S})}{r^2}\right) O(nlogn)$$

Example: vertical decomposition

Objects: line segments   Regions: Trapezoids with vertical lines   Conflict: a segments intersects a trapezoid.

The Algorithm

Data structures:

1. A list of trapezoids $\mathcal{D}ec_s(\mathcal{R})$. Each described by up to four segments and up to six edges.
2. Vertical Adjacency graph.
3. Conlict graph
    a. For each segment $S \in \mathcal{S}\backslash\mathcal{R}$, a list $\mathcal{L}(S)$ of conflicting trapezoids, from left to right.
    b. For each trapezoid $F \in \mathcal{D}ec_s(\mathcal{R})$ a list $\mathcal{L}'(F)$ of conflicting segments in $\mathcal{S}\backslash\mathcal{R}$.

Initialization

Build $\mathcal{D}ec_s(\mathcal{R})$ for one segment ($O(1)$), and create initial conflict graph ($O(n)$).

Updating the decomposition

Consider a new segment S. Traverse $\mathcal{L}(S)$ and split the trapezoid accordingly. Merging may be needed. Complexity of $O(n)$.

Updating the conflict graph

When a trapezoid F is split into F_i use traverse $\mathcal{L}'(F)$ and for each $S' \in \mathcal{L}'(F)$

1. Add S' to $\mathcal{L}'(F)$ is needed.
2. If $F \in \mathcal{L}(S')$, replace it by the F_i that intersect S'

If $F_1, \dots, F_k$ need to be merge, merge their lists $\mathcal{L}'(F_i)$, removing duplicates

Analysis

The algorithm obeys the update condition, hence the complexity is

$$O\left(n\sum_{r=1}^{n}\frac{f_0(r,\mathcal{S})}{r^2}\right)$$

Where $f_0(r,\mathcal{S})$ is the expected number of trapezoids of an r-sample.

Lemma 5.2.4: Given n segments with a intersections,

$$f_0(r,\mathcal{S}) = O\left(r + \frac{ar^2}{n^2}\right)$$

Online algorithms

The influence graph: a DAG where each node is associated with a region defined and without conflict at some point.

Property 1: The leaves correspond to regions defined and without conflict.

Property 2: The domain of influence of t node is contained in the domain of influence of its parents.

The algorithm

- Initial step: start from a minimal set $\mathcal{R}_0$ such that $\mathcal{F}_0(\mathcal{R}_0)$ is not empty. Create a fictitious root and connect each $F \in \mathcal{F}_0(\mathcal{R}_0)$ to it.
- Current step
    - Locating: Find all nodes that have a conflict with the new object O.
    - Updating: Find $\mathcal{F}(\mathcal{R} \cup \{O\})$. Create leaves for the each region created by O and connect to parents according to property 2.

Analysis

Theorem 5.3.2: The expected number of nodes in the influence graph is

$$v(\mathcal{S}) = O\left(\sum_{r=1}^{n} \frac{f_0(r, \mathcal{S})}{r}\right)$$

Update condition for influence graphs (5.3.3)

1. The existence of a conflict between a region and an object can be decided in constant time.
2. The number of children of each node is bounded by a constant.
3. The parents of a node created by O are the nodes that are killed by O and updating the influence graph takes time linear in the number of nodes killed or created at each step.

Theorem 5.3.4 (influence graph)

1. The expected storage to process n objects is

$$O\left(\sum_{r=1}^{n} \frac{f_0(r, \mathcal{S})}{r}\right)$$

נובע מ-5.3.2

2. The expected time complexity is

$$O\left(n \sum_{r=1}^{n} \frac{f_0(r, \mathcal{S})}{r^2}\right)$$

3. The expected time complexity of the locating phase at step k is

$$O\left(\sum_{r=1}^{k-1} \frac{f_0(r, \mathcal{S})}{r^2}\right)$$

4. The expected time complexity of the updating phase at step k is

$$O\left(\frac{f_0(k, \mathcal{S})}{k} + \frac{f_0(\lfloor (k-1)/2 \rfloor, \mathcal{S})}{k-1}\right)$$

Example: vertical decomposition

Each node contains the following:

1. A description of the corresponding trapezoid.
2. Up to four pointers for vertical adjacency.

At each step, for a new segment S:

1. Locate all the leaves/trapezoids that have a conflict with S, subdivide them and create the list $\mathcal{L}(S)$.
2. Traverse $\mathcal{L}(S)$ from left to right using the vertical adjacency. Create a node for each subtrapezoid and connect them to the node of the original trapezoid. If two subtrapezoids need to be merged, also merge their nodes.

Analysis

From theorem 5.2.4, $f_0(r) = O\left(r + \frac{ar^2}{n^2}\right)$. Hence the expected time complexity is $O(n \log n + a)$, the expected space complexity is $O(n + a)$ and the average time complexity for the n-th insertion is $O(\log n + a/n)$

Accelerated incremental algorithms

The idea: use a conflict graph to find the regions conflicting with S faster.

Theorem 5.4.1: knowledge of the conflict graph at step k can be used to perform the locating phase at a later step l with average complexity of

$$O\left(\sum_{r=k+1}^{l-1} \frac{f_0(\lfloor r/2 \rfloor, \mathcal{S})}{r^2}\right)$$

If $f_0(r) = O(r)$ then the complexity is $O\left(\log\left(\frac{l}{k}\right)\right)$.

Proof idea: For each object is the conflict graph, have pointers to the conflicting regions in the influence graph. At the locating phase, start from the k-th level of the influence graph, instead from the root.

Accelerated algoriothm: compute the conflict graph at steps $n_1, \dots, n_k, \dots$

Theorem 5.4.2: if $f_0(r, \mathcal{S}) = O(r)$ and the conflict graph at step k can be built in expected $O(n)$ time, then the randomized accelerated algorithm can run in expected time $O(n \log^* n)$