Overview

In the last lecture we opened the discussion about the Binary space partition trees (BSP). We saw some BSP applications, defined the size of BSP and calculated it for some special cases in $\mathbb{R}^2$.

In this lecture we will estimate an upper bound for the size of BSP tree segments in $\mathbb{R}^3$ and consider one more special case in $\mathbb{R}^2$ - bounded length segments.

Simple Upper Bounds for Segments in $\mathbb{R}^3$

*(Please, refer to Joe Mitchell’s slides on a paper with Adrian Dumitrescu and Micha Sharir, pages 20-25, which you can find on the class web page)*

We have a set $S$ of $n$ axis-parallel segments in $\mathbb{R}^3$. Let’s assume they are in *general position* for simplicity. That means no two segments share coordinates in any dimension.

*The picture is taken from the slides mentioned above*

**Theorem:** The complexity of BSP is $O(n^{3/2})$
Proof:

First, make $\sqrt{n}$ slices $\perp$ to $z$-axis

*The picture is taken from the slides mentioned above*

We can see from the picture, the groups of segments lie on separate disjoint planes, perpendicular to $z$-axis.

We can have $\leq \sqrt{n}$ red or green segments per slab.

Each black ($z$-parallel) segment is cut no more than $\sqrt{n}$ times.

This can be seen more clearly if we look at one of these planes in 2d projection.

*The picture is taken from the slides mentioned above*

We have $\leq \sqrt{n}$ horizontal/vertical segments and $\leq n$ black points on the $(x,y)$-plane.

Note also that each black segment is chopped into at most $\sqrt{n}$ pieces.

So, we have $\sqrt{n} \times \sqrt{n} = O(n)$ per slab.

Applying 2D BSP of size $O(n)$ per slice ($\sqrt{n}$ slices) leads to the conclusion $\rightarrow$ total size $O(n^{3/2})$
To summarize, we consider segments $E = X \cup Y \cup Z$ parallel to $x$-, $y$-, $z$-axes, where $x = |X|$, $y = |Y|$, $z = |Z|$, so that $x + y + z = n$.

**Theorem:** BSP size $\leq 4\sqrt{xyz} + 2n - z$, for $z \leq x, y$.

$$\leq \frac{4}{3\sqrt{3}}n^{3/2} + \frac{5}{3}n$$

$$\geq \frac{1}{3\sqrt{3}}n^{3/2} + n$$

**OPEN:** Close the gap in the constants

---

**Line Segments with Bounded length Ratios**

*(This part of the lecture is referred to the paper of de Berg, de Groot, and Overmars, pages 20-25, the link to it you can find on the class web page)*

Let $S$ be a set of $n$ disjoint line segments in the plane such that the ratio between the length of the longest segment and the length of the shortest one is bounded from above by a constant $c$.

**Theorem:** There exists a BSP of size $O(n)$, which can be constructed in time $O(n\log^2 n)$

**Proof:**

Our proof consists of two stages.

In the first stage we construct a partitioning such that each segment is intersected at least once (but not too many times). This implies that inside a cell of the partitioning all segments are connected to the boundary of that cell. On the next stage we construct for each cell a linear size BSP tree on the segments inside the cell.

**Stage I** To construct a BSP such that each segment is stabbed at least once we assume WLOG that the length of the shortest segment is one.

Next we add vertical and horizontal partition lines of the form $l(i/\sqrt{2})$:  

---

3
\[ y = \frac{i}{\sqrt{2}} \text{ horizontal partition lines} \]
\[ x = \frac{i}{\sqrt{2}} \text{ vertical partition lines} \]

Hence, we have the grid, which chops all the segments, whose length \( \geq 1 \). Any segment inside the cell of such a grid is \textit{anchored}, that is, it intersects the boundary of that cell.

**Lemma:** There exists a BSP that uses \((2c + 2)n - 2c + 2\) splitting lines, so that all the segments are \textit{anchored}.

**Stage II**

**Special subcase:** Let \( C \) be a convex cell. Let all segments be anchored at the same edge of \( C \). (This is called \textit{autopartition}).

**Lemma:** We can obtain a linear size BSP by taking splitting lines containing the segments, starting with the segment that extends farthest from the edge.

Sort the ends of the segments by the distance from the cutting edge. So, if we construct the BSP by taking the partition lines containing the segments according to this order, no segments will be fragmented. See the picture below.
General case: Now consider the case where the segments are incident to different edges of the boundary of the cell.

* (The picture is taken from the paper of de Berg I referred to in the beginning of this section)

Definition: The segment \( s_1 \) is defined to be a successor of \( s_0 \) if \( s_1 \) is the first segment hit by extending \( s_0 \).

We define the successor sequence of the segments \( s_0, s_1, s_2, ..., s_m \) so that \( s_1 \) is a successor of \( s_0 \), \( s_2 \) is a successor of \( s_1 \), etc., until the extension of one of the segments hits the boundary or one of the perviously extended segment forming a cycle.

Note that the process ends after \( \leq |S| \) steps, since a segment is added at most once to the sequence.

- **Case(i)** If extension of a segment hits the boundary, then take the partition lines containing the extended segments starting with the last one \( s_m \) and going backwards following the sequence order until we take \( s_0 \), recursing on the resulting subcells.

- **Case(ii)** Extension of a segment \( s_m \) hits the extension of some segment \( s_i \).

Now we have a cycle \( s_i, ..., s_m \) and we need to “break” it by extending on of the segments even further. Then we can use that segment to start the partitioning process in the same way as we did for the case (i). However, extending one of the segments may cause a lot of fragmentation, which we can not control. Here we need to process in a slightly different fashion.

Let \( p_i \) be the endpoint of the extension of \( s_i \) that is on the boundary of the cell and \( q_i \) be the other endpoint (see the picture below). First take as a partition line the segment connecting \( p_m \) and \( p_{m-1} \). Next we extend \( s_{m-1} \) until it hits \( p_m p_{m-1} \) and add this extension as a partition line. We have now broken a cycle and we can add the extensions of \( s_{m-3} , s_{m-4} ... s_0 \) as partition lines. This is illustrated in picture (i) below.
* (The picture is taken from the paper of de Berg I referred to in the beginning of this section)

The partition lines can cut segments into fragments. All the cut segments lie inside at most three subcells, which lie inside the convex region shaded on the picture (i). Recursively we can partition them to get a valid BSP. Note that triangle $p_{m-1}q_{m-1}p_m$ has the only one edge which cuts the segments. Hence, we can construct a BSP for the segments inside triangle using the special subcase we considered above.

The resulting partitioning is illustrated on the picture (ii) above.

Summarizing, we obtained a linear size BSP for a set of segments in the plane with bounded length ratios in the following way. First we constructed a partitioning such that each segment is intersected at least once. Then we solved the subproblem inside each cell using the method for anchored segments described in Lecture 1.

Hence, we obtained a BSP of size $O(n)$ in time $O(n \log^2 n)$ using $O(n)$ space.