# Randomized Parallel Algorithms
for
# Trapezoidal Diagrams


| Ken Clarkson | Richard Cole | Robert E. Tarjan |
|---|---|---|
| AT&T Bell Labs | Courant Inst. | Princeton Univ., |
| Murray Hill, NJ | New York Univ. | NEC Research Inst. |

# Outline

- Trapezoidal Diagrams
- Results
- Ideas
- Conclusions

# Trapezoidal Diagrams

Given set $S$ of
$n$ line segments, with
$A$ intersection points,
its TD $\mathcal{T}(S)$ has $\Theta(n + A)$ regions.

Results

Suppose $S$ forms $K$ known chains.
How much work is needed to find $\mathcal{T}(S)$,
and how quickly can the diagram be found?

| | work | time/ $\lg n$ | |
|---|---|---|---|
| $\Omega(K \lg n + A$ | $+n)$ | $\Omega(1)$ | |
| $K \lg n + A$ | $+n \lg^* n$ | $\lg \lg n \lg^* n$ | |
| $A$ | $+n \lg n$ | $1$ | |
| | $n$ | $\lg \lg n \lg^* n$ | simple |
| | | | |
| | $n$ | • | simple; C |
| | $n \lg^* n$ | • | CTVW, S |
| | | | |
| | $n^2$ | $1$ | HJW |
| | $n^2$ | $\lg^* n$ | AM, (G) |
| $A \lg n$ | $+n \lg^2 n$ | $1$ | Goodrich |
| $A$ | $+n \lg n$ | $1$ | red/blue; GSG |

## The Model

Expected work and worst-case time
implies processors are expected?
CREW PRAM,
processor allocation every $\log n$ steps

Randomized divide-and-conquer [CS]:

- take $R \subset S$ random of size $r$;

- compute $\mathcal{T}(R)$;

- for $T \in \mathcal{T}(R)$, find segments $S_T$ meeting it (insertion);

- compute $T \cap \mathcal{T}(S_T)$ for $T \in \mathcal{T}(R)$;

- merge pieces to find $\mathcal{T}(S)$;

We can use “slow” algorithms
for $\mathcal{T}(R)$ and the $T \cap \mathcal{T}(S_T)$, since:

Each trapezoid meets
$O(n/r)$ segments, on average, and
$O(n/r) \log r$ with high probability.

For parallel work $O(A + n \log n)$,
use Goodrich’s algorithm to compute $\mathcal{T}(R)$,
and a quadratic algorithm like [HJW] for sub-problems.

Serially, for simple chains:
to insert, walk through $\mathcal{T}(R)$ and $S$;

This gives $O(n \log\log n)$ expected time,
with $r = n/\log n$ and average subproblem size $O(\log n)$.

For $O(n \log^* n)$ work:
For subsets
$S^1 \subset S^2 \subset \cdots \subset S^{\log^* n} = S$,
with $|S^1| = n/\log n$, $|S^2| = n/\log\log n$,
$|S^i| = n/\log^{(i)} n$,
compute $\mathcal{T}(S^i)$ using $\mathcal{T}(S^{i-1})$.

In parallel, the insertion is done by
many parallel walks through subchains.

The main problem: while every trapezoid of $\mathcal{T}(R)$ meets few segments,

*a segment may meet many trapezoids.*

How to handle *bad* segments
that meet $\Omega(\log n)$ trapezoids?

There are $O(n/\log n)$ bad segments,
on average: to insert them,
compute their intersections
with the visibility edges using algorithm [GSG].

# Conclusions

- realistic machine models;
- determinism;
- simple $O(n)$ triangulation?