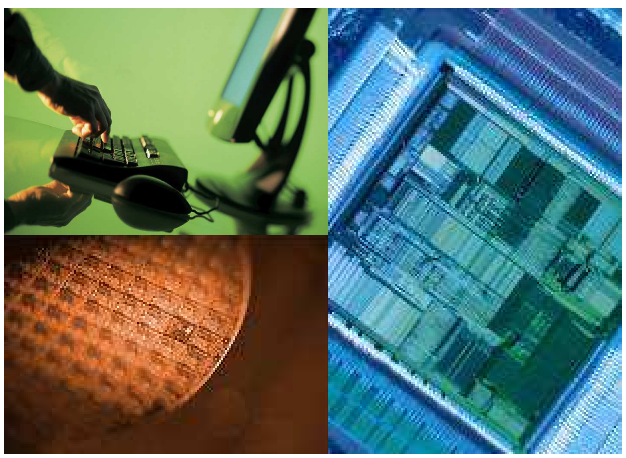
An Innovative Steiner Tree Based Approach for Polygon Partitioning



Yongqiang Lv
Qing Su
Jamil Kawa

January 23, 2008

SYNOPSYS
Predictable Success

Outline

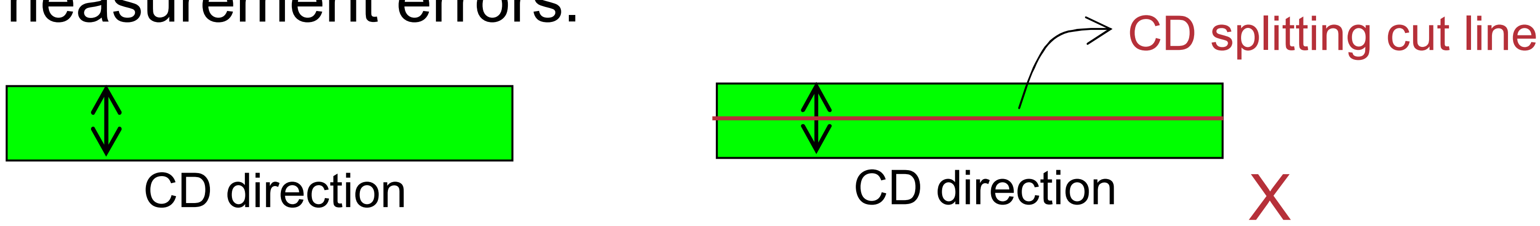
- Motivation.
- Quality issues with polygon partitioning in MDP.
- Existing work.
- Our proposed approach.
  - Overview of proposed solution: MPT.
  - Optimization objective.
  - MPT formulation.
  - MPT construction algorithm.
- Simulation results.
- Conclusions.

Motivation

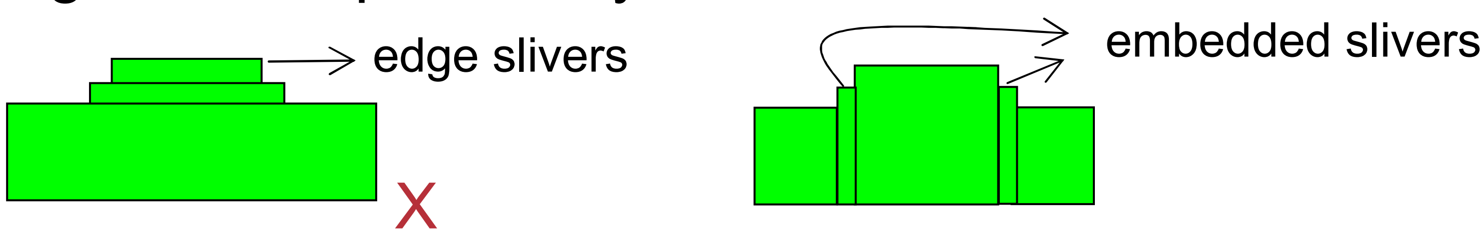
- Impact of technology scaling down on MDP:
  - More complicated RET lead to more complex MDP.
  - Run time and quality issues become more severe.
- Polygon partitioning is a key step in MDP.
  - Decompose the polygon into non-overlapping axis parallel trapezoids with rectilinear cut lines.
    - Arbitrary polygons = union of disjoint x-traps and y-traps.
  - Quality requirements are growing more stringent due to tighter CDs.
    - More on next slide.
- Modern MDP requires polygon partitioning algorithm to have:
  - High quality of results, capable of handling tighter CDs induced quality metrics;
  - Low computational cost.

Main Quality Requirements for Polygon Partitioning

- Smaller partitioning figure count ➔ to reduce mask writing time;
- No CD-splitting cut lines ➔ to avoid additional CD measurement errors.



- Minimized sliver count and sliver edge length ➔ to guarantee printability.



Existing Work

- Many published work for general polygon decomposition/partitioning.
  - Mainly for the purpose of meshing/triangulation.
  - Most are not applicable for MDP because mask writing tools requires rectilinear cut lines.
- Polygon partitioning algorithms for MDP application:
  - Minimize partitioning figure count.
    - Bi-partite based methods.
      - Partition the polygon into sub-polygons that does not contain chords.
      - Hard to address cut quality issues.

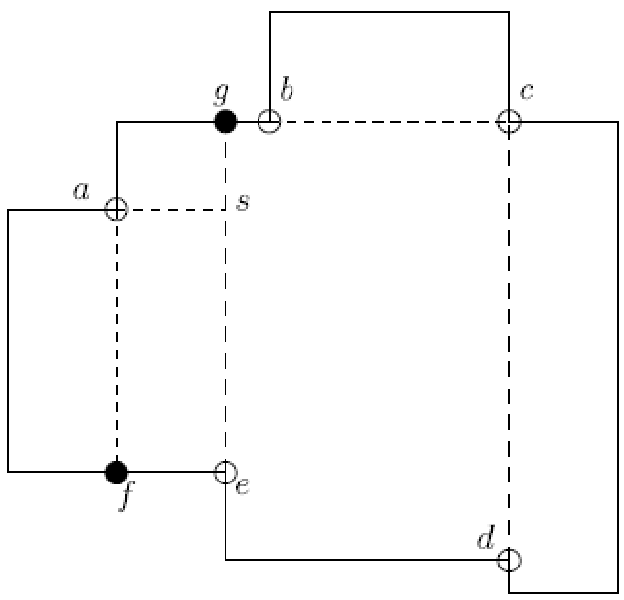
Existing Work (cont'd)

- Polygon partitioning algorithms for MDP application:
  - Minimize partitioning figure count.
  - **Optimize the partitioning figure shape. (Our focus.)**
    - Objective: mainly "min cut line length", some work uses "max min cut line".
      - Benefit of using this objective: has better control on cut quality.
      - Our approach can apply to both objectives.
      - Can combine with "min figure count" objective as well.
    - Most methods process cut line one by one (we refer to as "cut-line based")
      - Processing order is critical. Need iterations of evaluate-modify/re-evaluate cycles.
    - Some work is based on recursively splitting sub-polygons ($O(n^4)$).
    - Some work uses ILP based method to work on grid edges and vertices formed by all candidate cut lines.

Our Solution

- **A totally different approach.**
  - Objective: minimize cut line length.
    - Equivalent to making partitioning figures have aspect ratio close to 1.
  - Minimal partition tree (MPT) algorithm based on Steiner tree.
    - Cut lines in the optimal partition are obtained from optimal Steiner trees' edges.
  - Reformulate the tree edge cost and constraints to include quality metrics.
  - Great QoR, with low run time (O(nlogn+Cn)).
- **Key advantages.**
  - Optimization of all the cut lines globally by construction.
    - No order dependency. Avoids evaluate-modify-re-evaluate cycles.
  - Many mature and efficient Steiner tree algorithms can be utilized by this approach.
    - Good quality and fast runtime.
  - Changing optimization objectives become easy.

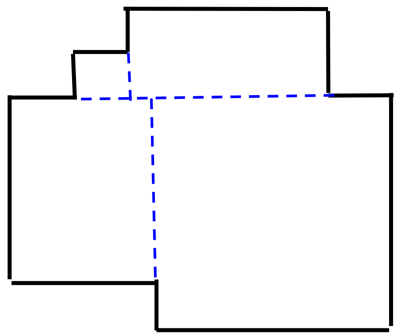
Some Terminologies



- Inflection vertex (I-vertex)
  example: a, b, c, d, e
- Cut lines
  example: bc, cd, eg, af
  - chord vs. cutting ray
    example: bc, cd are chords
    eg, af are cutting rays
- Ray crossing vertex (R-vertex)
  example: f, g
- R-vertex's parent I-vertex
  example: e is g's parent I-vertex
  a is f's parent I-vertex
- Partition graph
  - Formed by I-vertex, R-vertex, and cut lines.
  - Cost is defined as edge (cut line) length.
  - Incorporates sliver cost, CD-slicing cost.

Objective: Minimum Cut-line Length Rectilinear Partition (MCLRP)

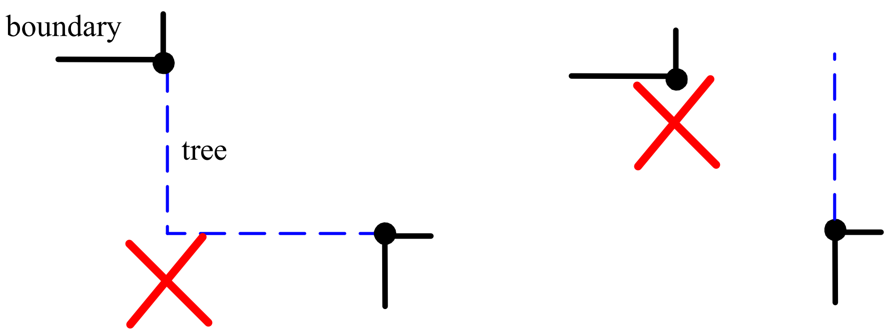
- MCLRP: minimum total cut-line length rectilinear partition



- MCLRP is proved to be equivalent to be the partition with partitioning figures' aspect ratio closest to 1.
  - Partitioning figures are closer to square, rather than low narrow rectangles.
- One important property of MCLRP: each inflection vertex has 1 and only 1 cut line, unless in chord case.
  - It is used as a constraint for our variant Steiner tree.

Minimal Partition Tree (MPT) Formulation

- Use all the I-vertices and R-vertices as tree terminals
- Build Steiner minimal tree (total edge cost minimized)
  - Cost of tree edge: the non-boundary portion of the L1 distance between two end points.
  - Tree edges forming slivers or CD slicing have high cost.
- Special constraints required by this variant Steiner tree:
  - No L-shape edges;
  - Each I-vertex must have least one edge in cut-state.

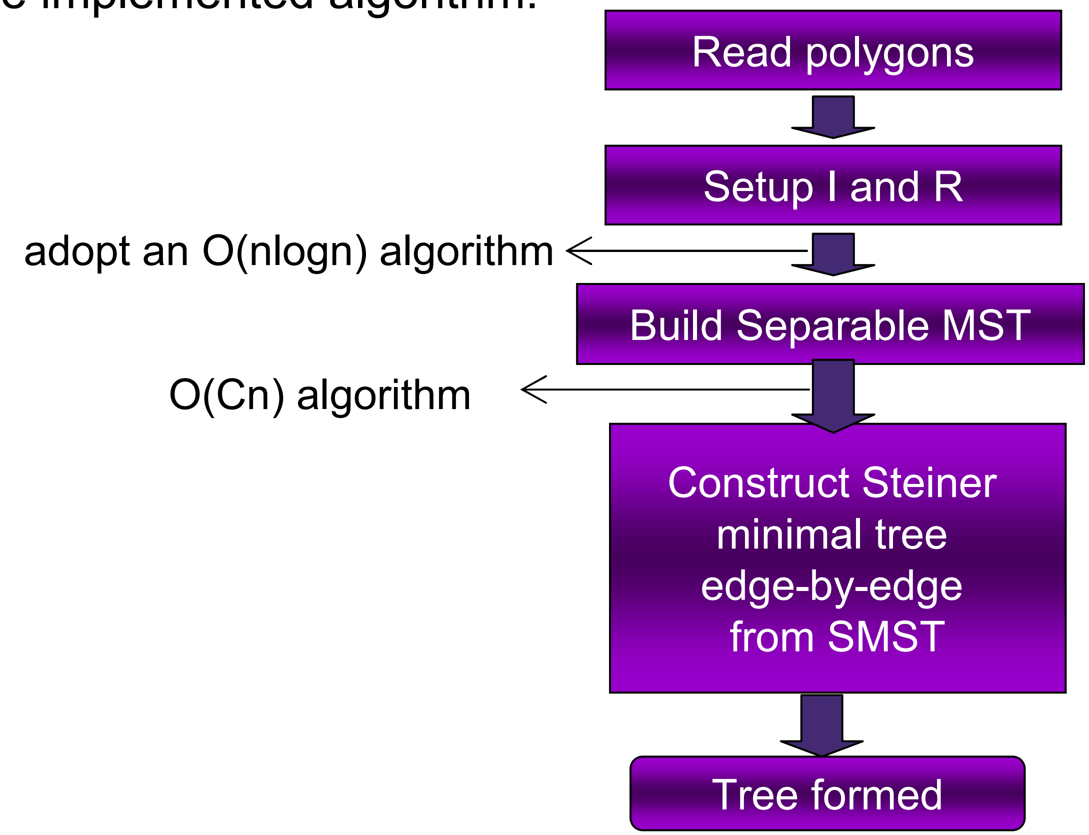


(a) No L-shape (b) No un-cut

- Ultimately we obtain a minimal cut line length partition MCLRP, with sliver and CD slicing controls.

MPT Construction Algorithm

- Any Steiner tree construction algorithm can serve as basis for MPT construction.
  - As long as the tree is constructed edge by edge.
  - The edge cost formulation can easily include the sliver/CD metrics.
- One implemented algorithm:



Examples

- Tested on 18 real design examples and many randomly generated examples.
  - 10^3 – 10^6 polygon vertices.
- List 5 Examples:
  - p3 and p5 are randomly generated.
  - ex02, ex05, ex12 are real design examples.

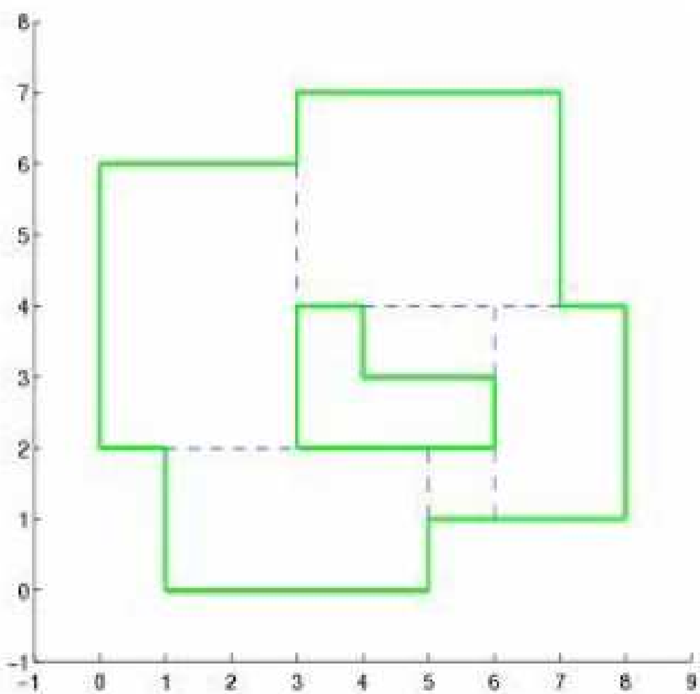
Example	#polygon	#hole	#vertex	#I-vertex	Sliver size
p3	1	1	18	9	1
p5	1	0	26	11	1
ex02	1	1470	18260	12068	100nm
ex05	7	0	792	382	100nm
ex12	421	0	19524	9070	100nm

Results

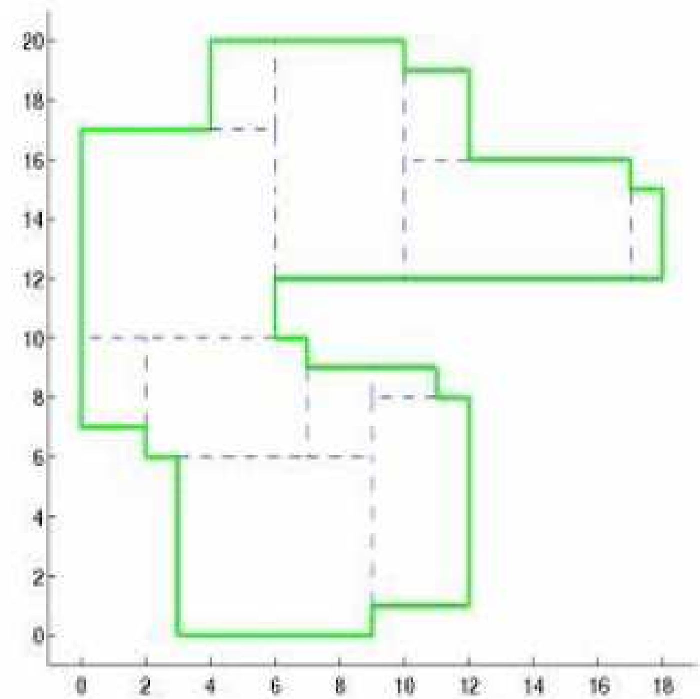
- Better quality, low run time complexity.
- Partitioning results – statistics.

Example	#rectangle	#slivers	#edge sliver	CPU (s)
p3	6	2	0	0
p5	12	2	2	0
ex02	7568	35 (unavoidable)	35	3
ex05	343	83 (embedded)	6	0.2
ex12	8792	19 (embedded)	1	0.5

Snapshots of Two Examples



(a) p3



(b) p5

Conclusions

- Introduced **a new approach** for polygon partitioning, based on Steiner tree construction.
  - Globally evaluate all candidate cut lines by using tree-construction algorithm.
  - Make use of mature algorithms from Steiner tree research.
  - Under the proposed framework, changing optimization objectives is easy.
- Proposed a way of **formulating** variant Steiner trees which can lead us to MCLRP partition.
- Proposed an **algorithm** constructing the formulated variant Steiner tree (referred to as minimal partition tree).
  - The method provides high quality of result with low run time complexity.
- This work also provides the **theoretical and algorithmic foundation** for applying other Steiner tree algorithms to polygon partitioning problems.