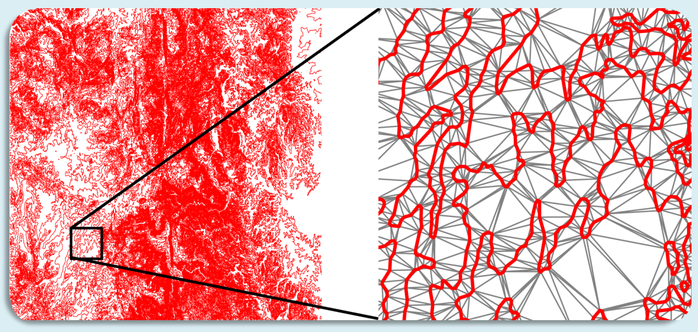
Delaunay Mesh Generation using the GPU

http://geomGPU.net

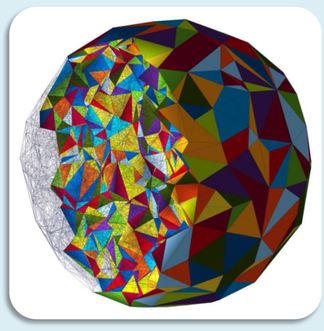
Thanh-Tung Cao, Mingcen Gao, Meng Qi, Tiow-Seng Tan
School of Computing, National University of Singapore

Ashwin Nanjappa
Bioinformatics Institute, Singapore

Zhiyong Huang
Institute for Infocomm Research, A*STAR



Contour map

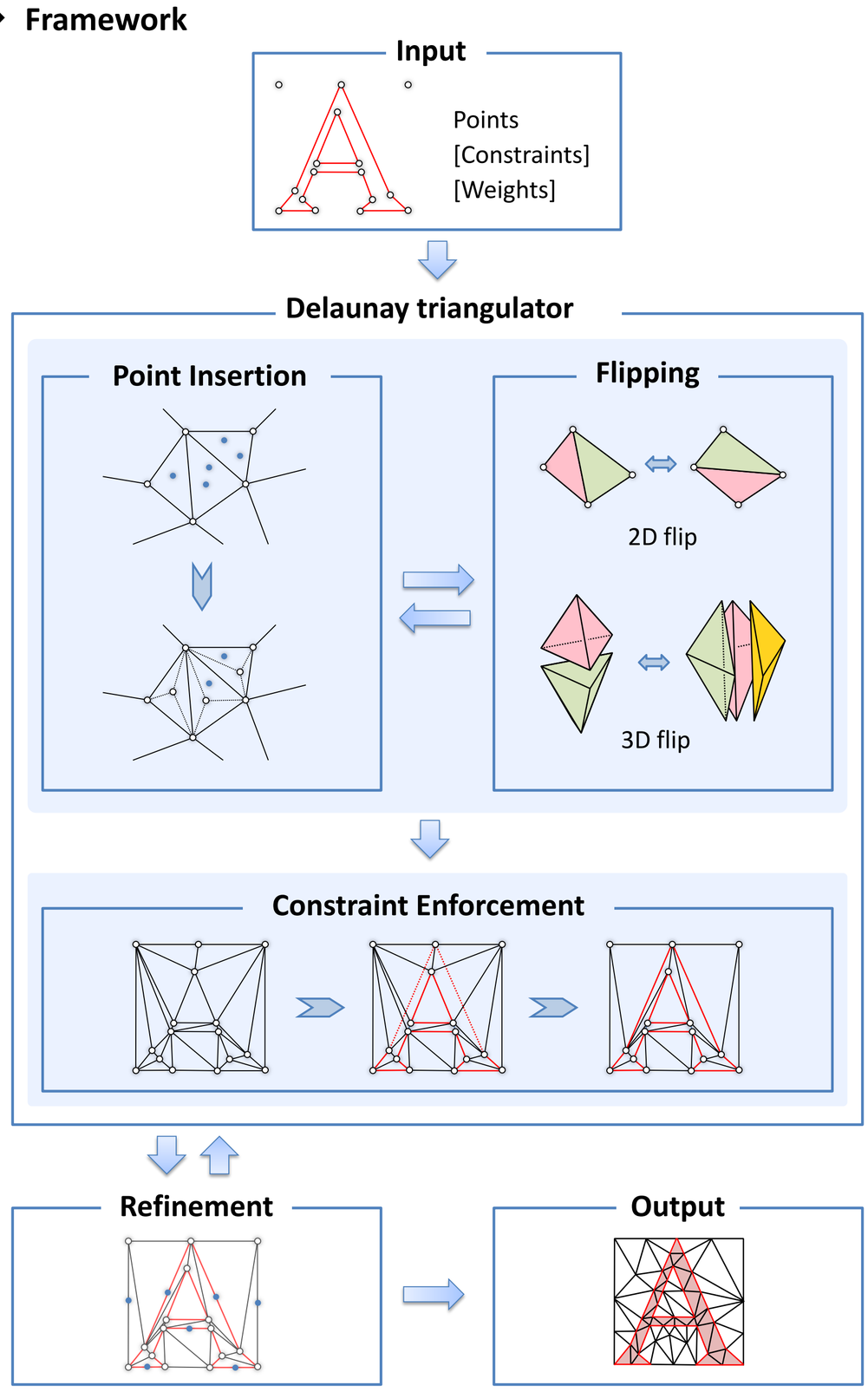


2D Constrained Delaunay triangulation

3D Delaunay triangulation

❖ **Problem**: Generate high-quality Delaunay mesh using the GPU.
❖ **Input**: Point set in $\mathbb{R}^2$ or $\mathbb{R}^3$ with [constraints] and [point weights].

❖ Framework

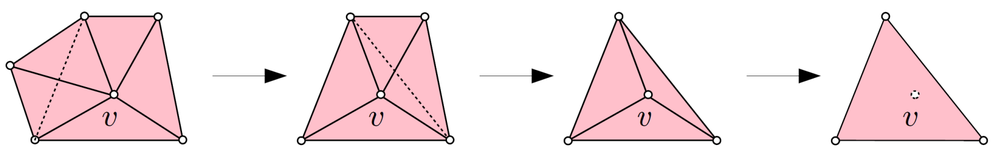


❖ 2D Delaunay Triangulation [1]

A direct application of our framework with any flipping sequence efficiently leads to the 2D Delaunay triangulation.

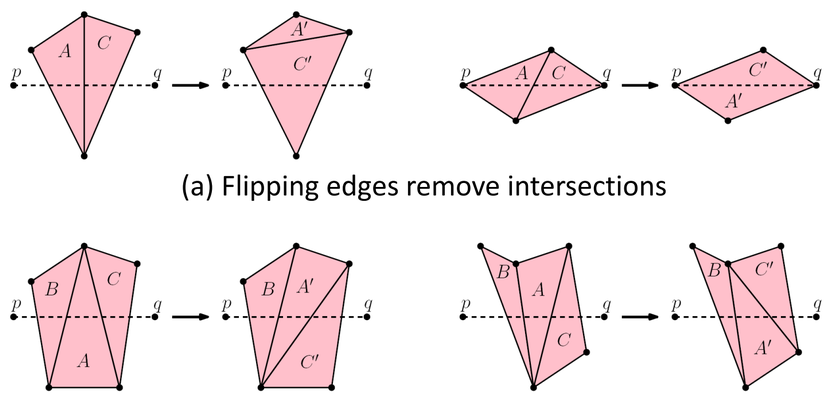
❖ 2D Regular Triangulation [2]

- *Key techniques*:
  - Identify redundant points through unflippable edges.
  - Combine flipping of regular and non-regular edges to remove redundant points.



❖ 2D Constrained Delaunay Triangulation [1]

- Constraint enforcement is performed by repeating two steps:
  - Identify triangles intersected by constraints.
  - Flip edges to remove intersections.
- *Key techniques*:

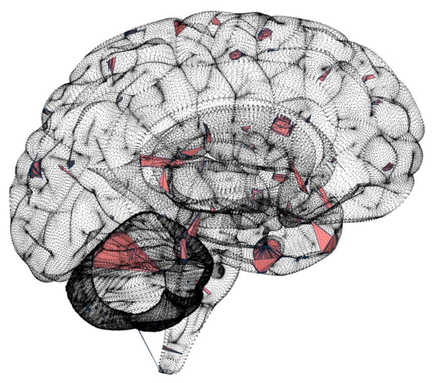


(a) Flipping edges remove intersections

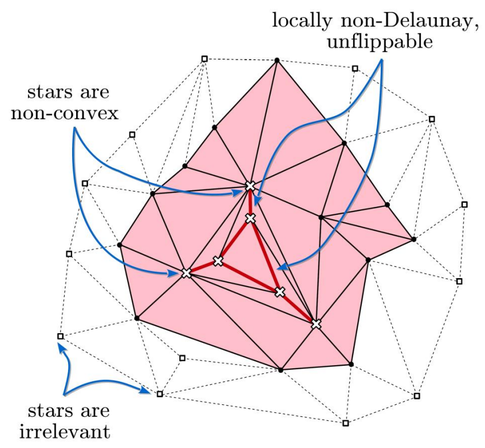
(b) One step look-ahead

❖ 3D Delaunay Triangulation [3]

- *Key techniques*:

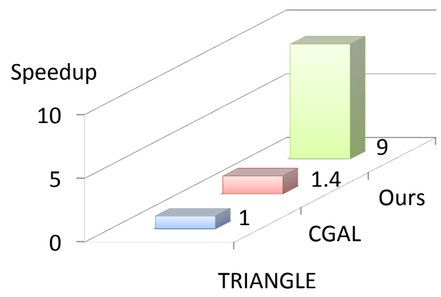


(a) After flipping, only a few regions contain locally non-Delaunay facets.

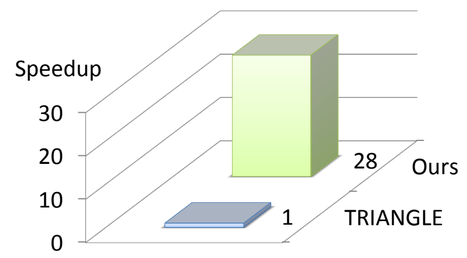


(b) Adaptive star splaying locally fix these regions.

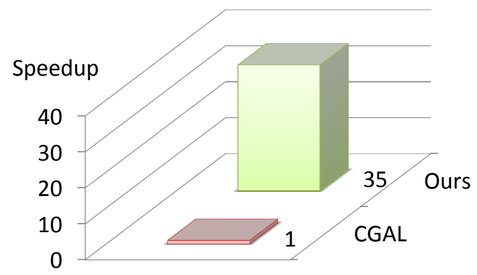
❖ Performance



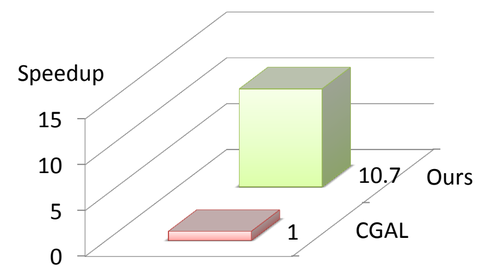
2D Delaunay triangulation



2D constrained Delaunay triangulation



2D regular triangulation



3D Delaunay triangulation

❖ Ongoing work

- Constrained Delaunay triangulation in $\mathbb{R}^3$.
- Delaunay refinement in $\mathbb{R}^2$ and $\mathbb{R}^3$.
- Local transformation in higher dimensions.

❖ References

[1] *Computing 2D constrained Delaunay triangulation using the GPU* [TVCG '12]
[2] *Flip-Flop: Convex hull construction via star-shaped polyhedron in 3D* [i3D '13]
[3] *A GPU accelerated algorithm for 3D Delaunay triangulation* [i3D '14]