Flip-Flop: Convex Hull Construction via Star-Shaped Polyhedron in 3D

Mingcen Gao Thanh-Tung Cao Tiow-Seng Tan Zhiyong Huang

G^3 Lab, School of Computing, National University of Singapore

Institute for Infocomm Research, A*STAR

1 Contribution

- ***Flip-Flop***: a novel flip algorithm to transform any star-shaped polyhedron to its convex hull.
- ***ffHull***: an algorithm to construct 3D convex hull using flips that works well on both CPUs and GPUs.
- ***ffRT***: the first known algorithm to construct the 2D regular triangulation on the GPU.

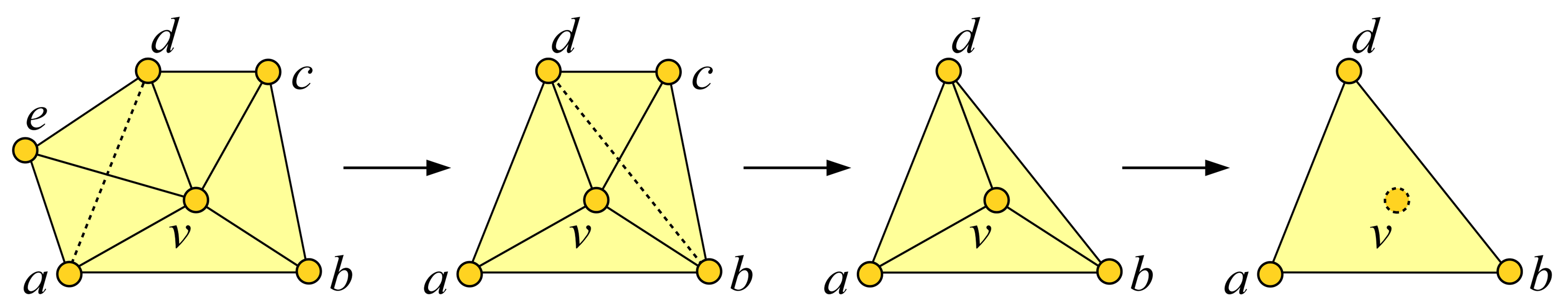
3 Flip-Flop: a novel flip algorithm

V-criterion: increases the **V**olume of the polyhedron by flipping the reflex edges

→ used in the traditional flip algorithm

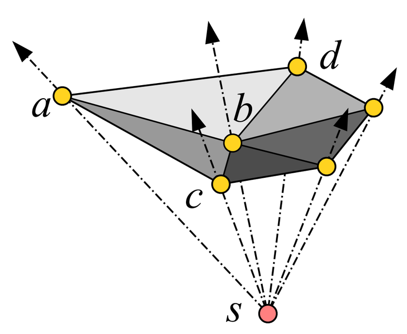
D-criterion: decreases the **D**egree of the non-extreme points by flipping the edges incident to them

→ a new criterion to remove non-extreme points

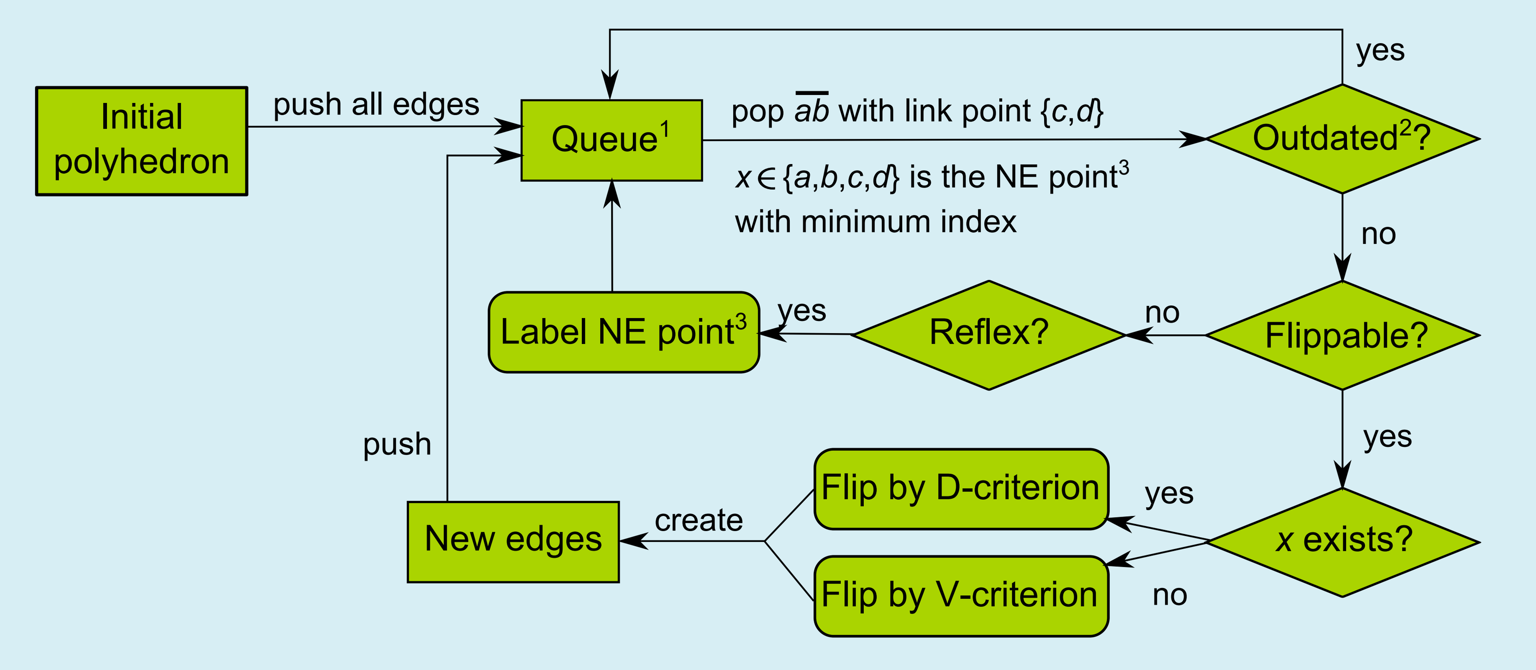


Finding non-extreme points:

a reflex edge that is unflippable implies a non-extreme point

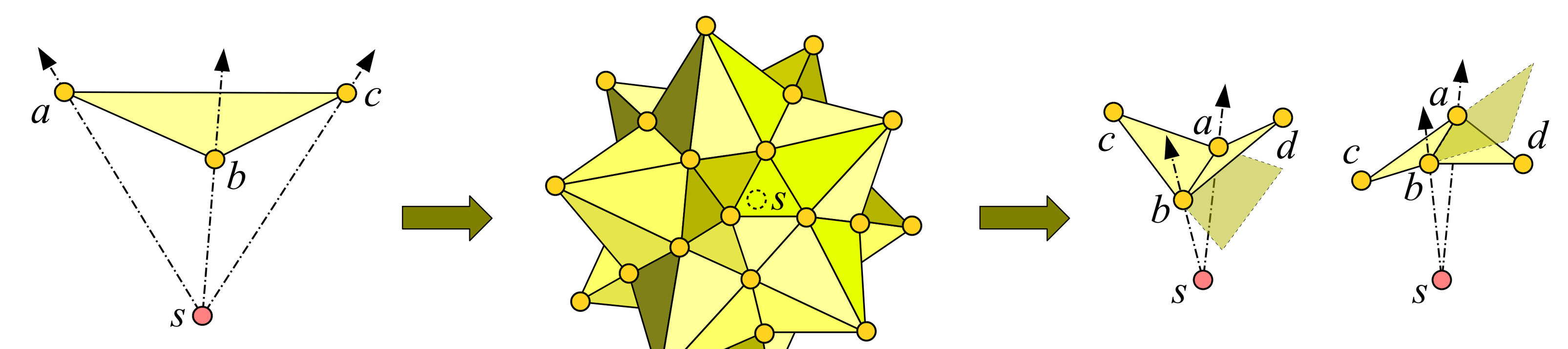


Flip-Flop: combines V- and D-criterion to compute convex hull



1: The Flip-Flop algorithm terminates when the queue is empty.
2: $\overline{ab}$ is outdated if it no longer exists or its link points have changed.
3: NE = non-extreme.

2 Flipping on Star-shaped Polyhedron

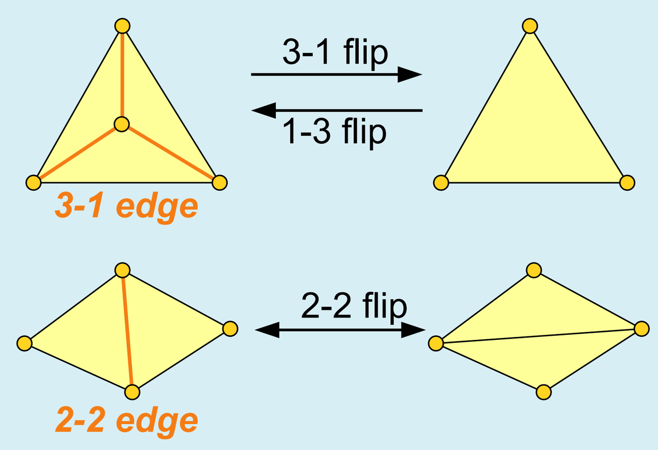


the **cone** of Δabc w.r.t. s, $C_s(\Delta abc)$, is the convex hull of $\{\overrightarrow{sa}, \overrightarrow{sb}, \overrightarrow{sc}\}$.

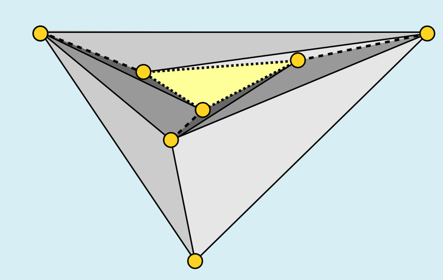
A polyhedron is ***star-shaped*** w.r.t. s if the cones of the triangles w.r.t s do not overlap.

Edge $\overline{ab}$ is ***reflex*** if d and s lie on different sides of Δabc. Otherwise it is ***convex***.

Topological flipping on 2D triangulation

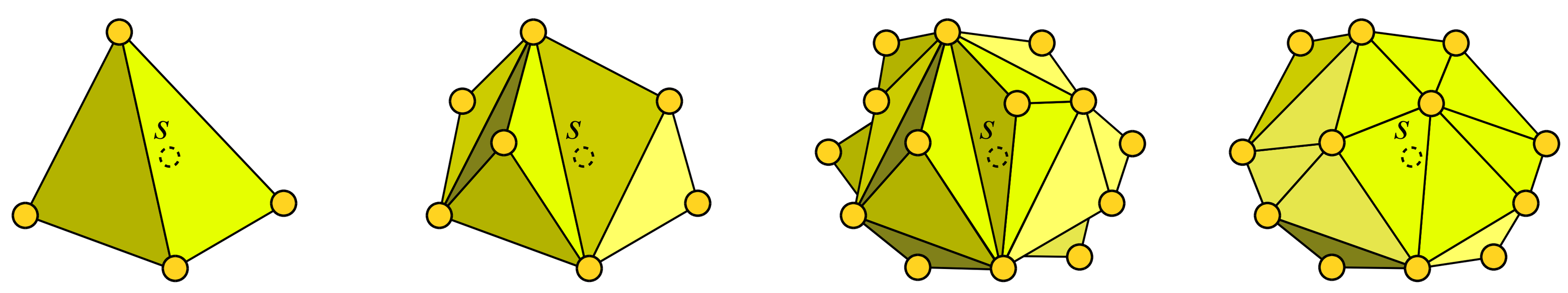


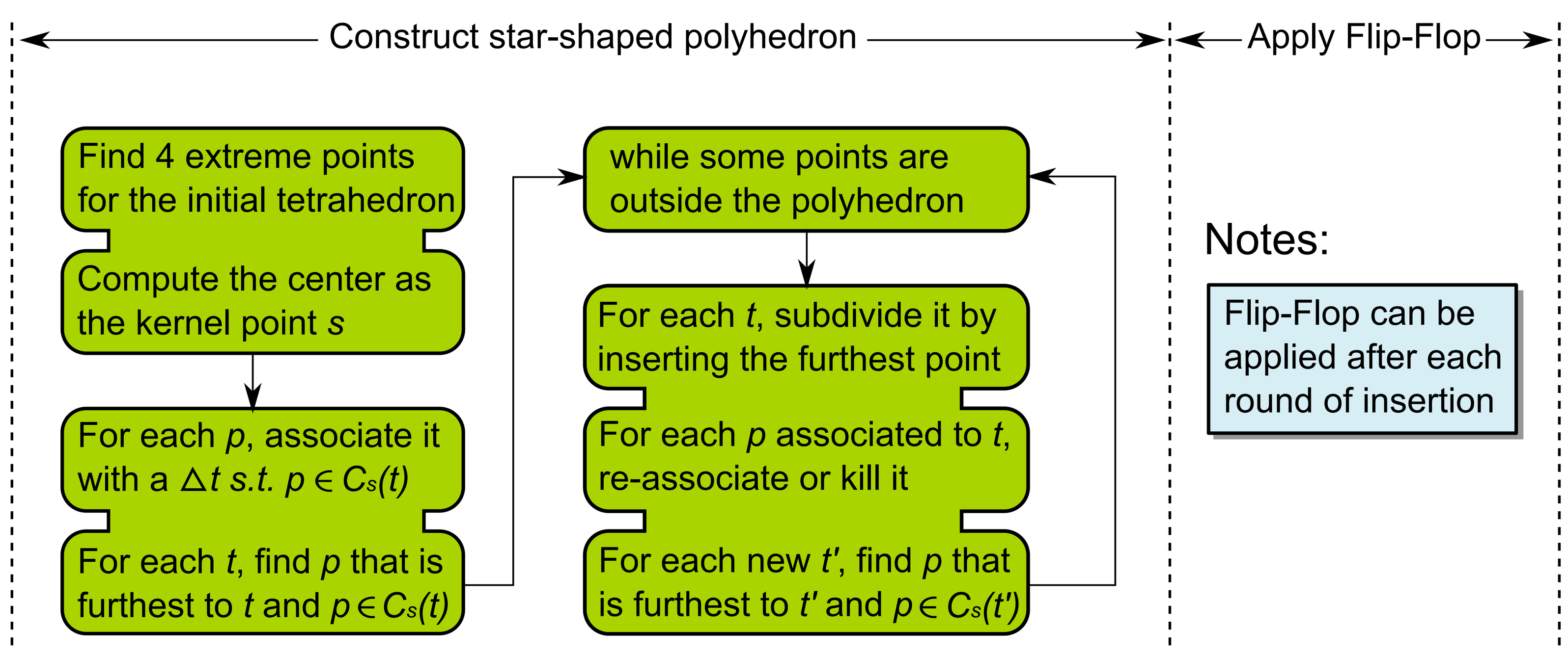
Tranditional flip algorithm may get stuck



The dashed edges are reflex but ***unflippable*** (i.e. flipping them creates self-intersection)

4 ffHull: 3D convex hull algorithm





Notes:

Flip-Flop can be applied after each round of insertion

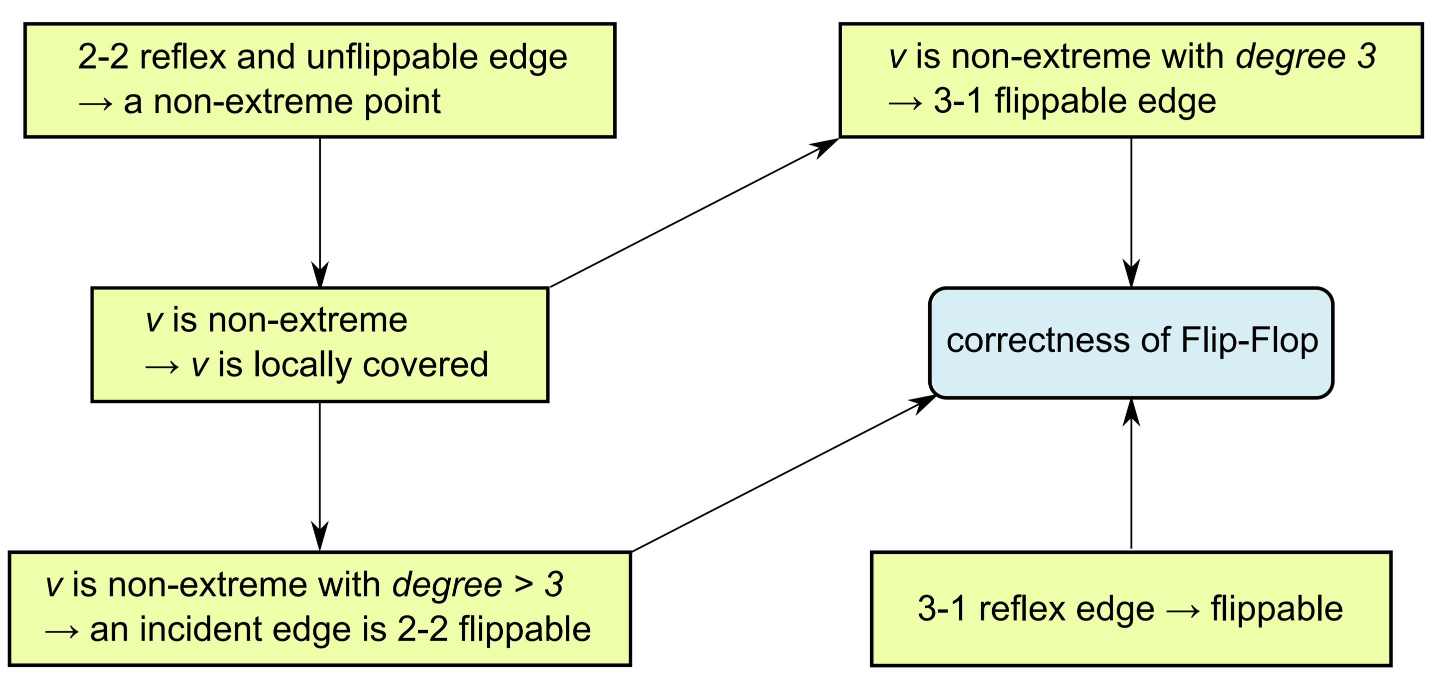
Implementation Tips:

- Maintain the orientation of each triangle so that the kernel point s is always beneath it.
- Use orientation determinants when finding the furthest point.
- In GPU implementation, seperate the kernels for fast and exact predicates.

5 ffRT: 2D regular triangulation algorithm

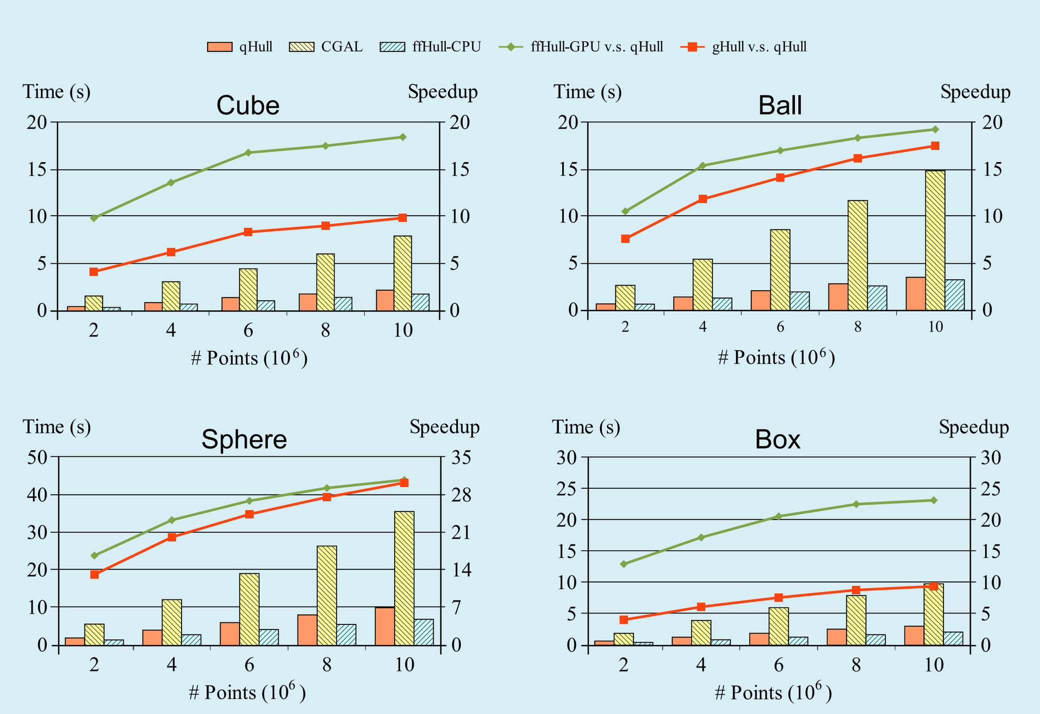
A set of weighted points → Lift to 3D space by $z=x^2+y^2-w$ → Compute the lower hull using ffHull → RT

6 Correctness of Flip-Flop



7 Experiments

Convex Hull



Regular Triangulation