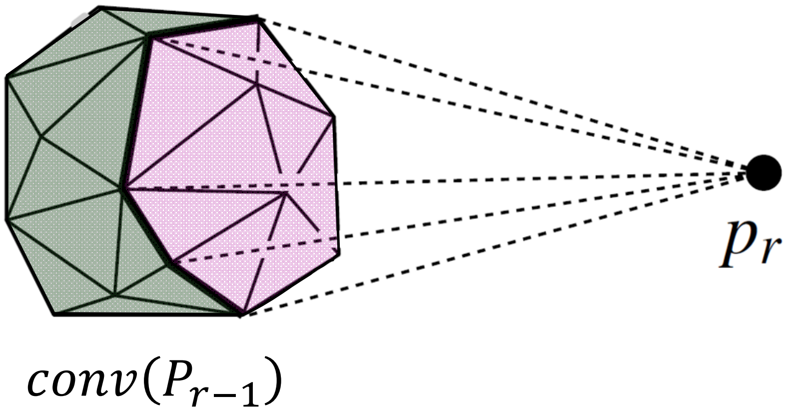
CMPS 6640/4040 Computational Geometry
Spring 2016

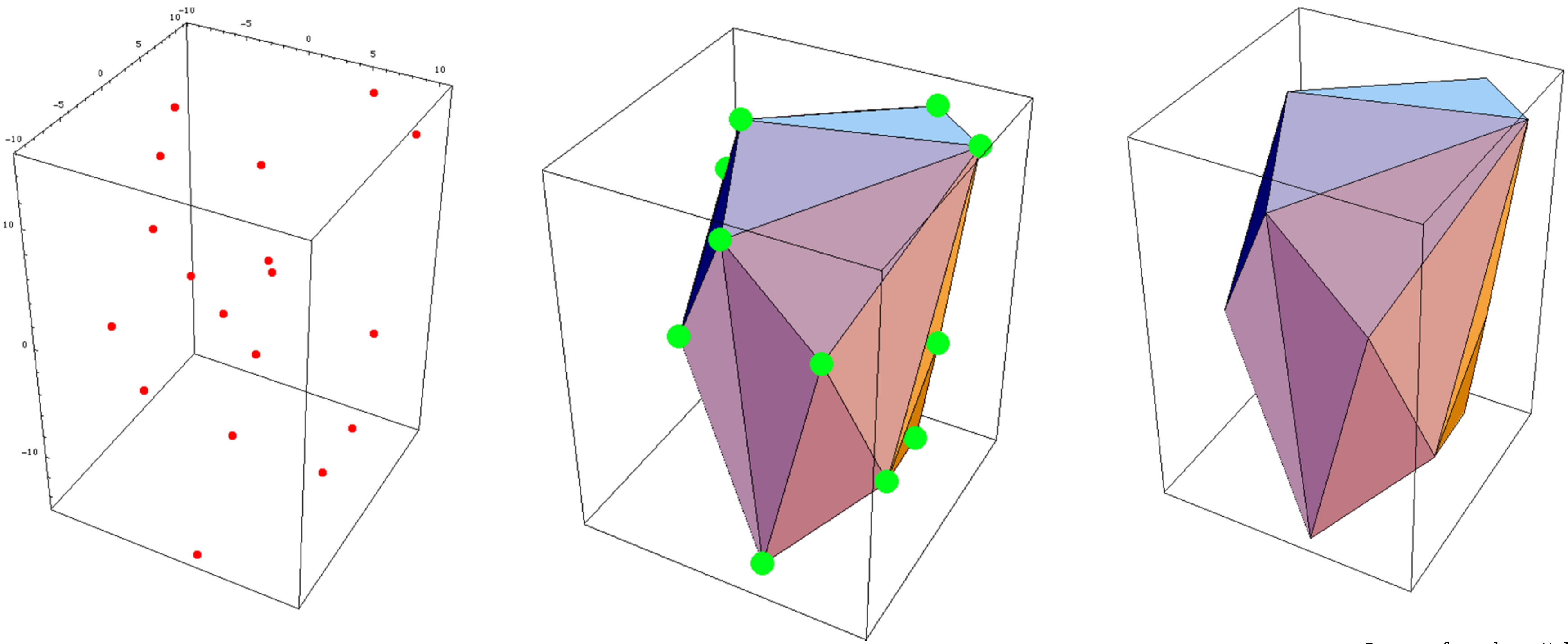


3D Convex Hulls

Carola Wenk

(based on BCKO)

3D CH: Problem Statement



Images from http://xlr8r.info/

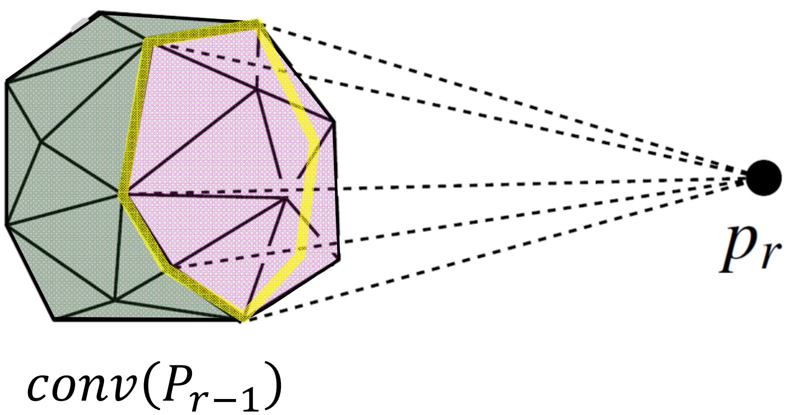
- Given a set $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^3$, compute $conv(P)$
- Use a DCEL to represent the boundary $\partial\ conv(P)$

Clarkson & Shor's Randomized Incremental Construction (RIC)

1. Choose 4 points in P that do not lie in a common plane. (Otherwise apply planar CH algorithm.)
   Wlog, let these points be p_1, p_2, p_3, p_4
2. Let $p_5, \ldots, p_n$ be a random permutation of the remaining points in P.
3. Define $P_r = \{p_1, \ldots, p_r\}$ for $r \geq 1$
   for $r = 5; r \leq n; r$**++**
   Compute $conv(P_r)$ by inserting p_r into $conv(P_{r-1})$

Visible and Invisible Regions

- If $p_r \in conv(P_{r-1})$, then conv(P_r) $= conv(P_{r-1})$
- Now, consider the other case $p_r \notin conv(P_{r-1})$

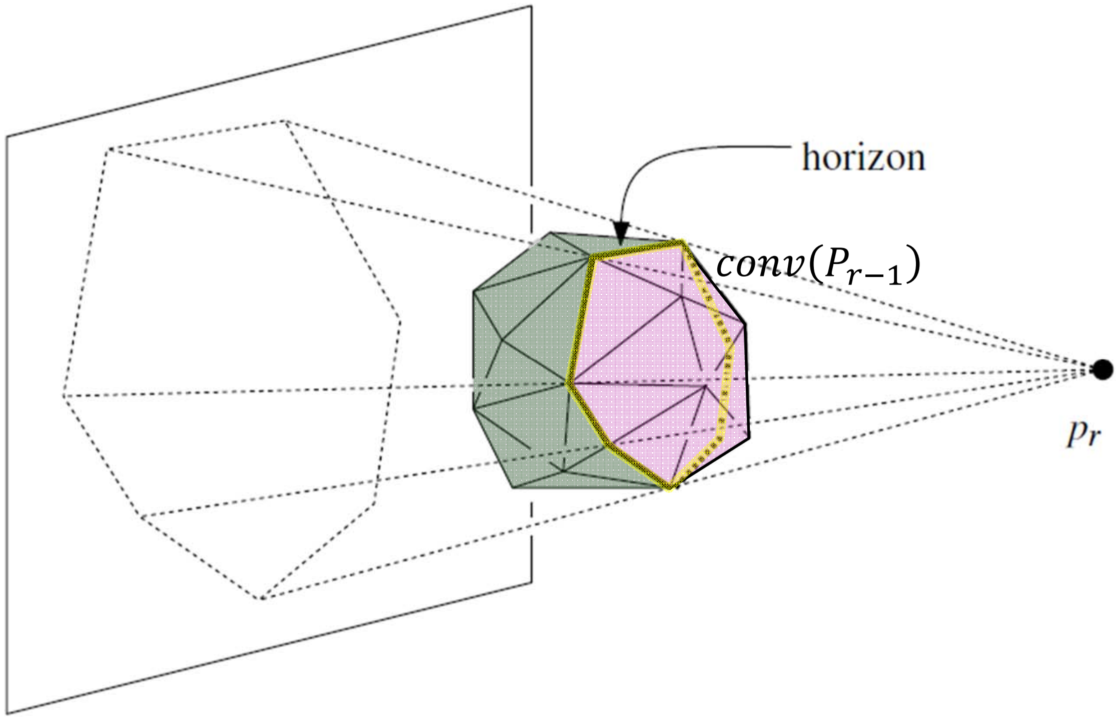


Look at $conv(P_{r-1})$ from p_r
$\Rightarrow$ *visible region* and
invisible region

The visible region and invisible region are connected regions on $conv(P_{r-1})$, separated by the *horizon*.

The horizon is a closed curve consisting of edges on $conv(P_{r-1})$.

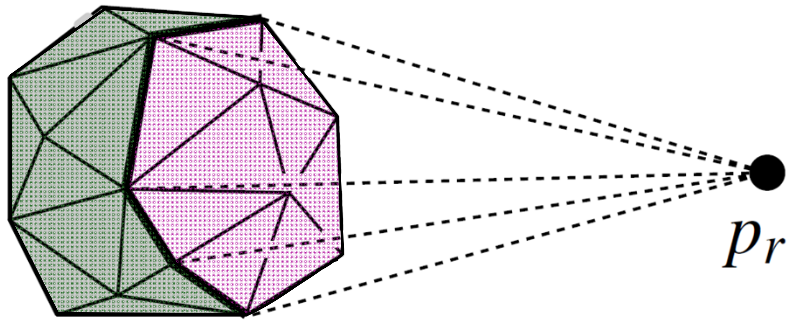
Horizon



Project $conv(P_{r-1})$ onto a plane with p_r as the center of projection.

$\Rightarrow$ Convex polygon that "equals" the horizon.

Visibility

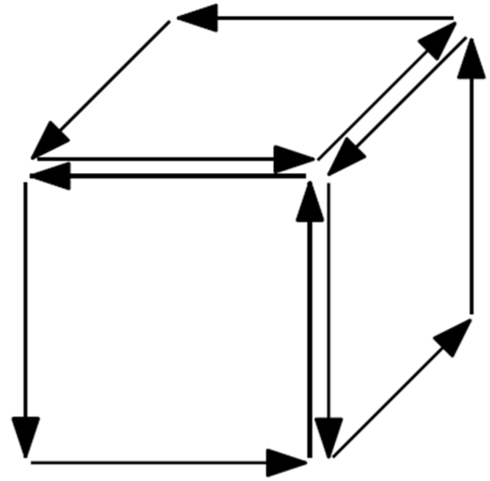


A facet f on $conv(P_{r-1})$ is *visible* from p_r
$:\Leftrightarrow p_r$ lies in h_f^+, where h_f is the plane containing f, and h_f^+ is the open halfspace that does not contain $conv(P_{r-1})$

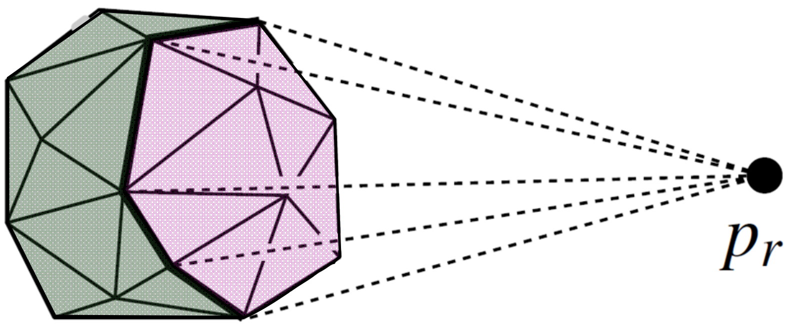
Storage Data Structure

Store the boundary of $conv(P)$, and of all intermediate $conv(P_r)$, as a DCEL. The vertices are 3D points.

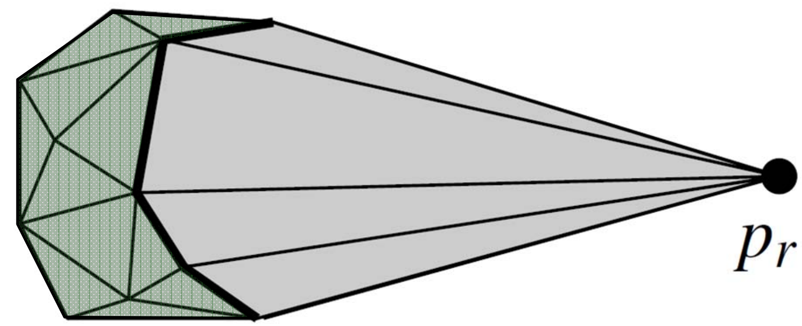
Wlog, half-edges bounding a face that is seen from the outside of the polytope form a counter-clockwise cycle.



Compute $conv(P_r)$ **from** $conv(P_{r-1})$



$conv(P_{r-1})$



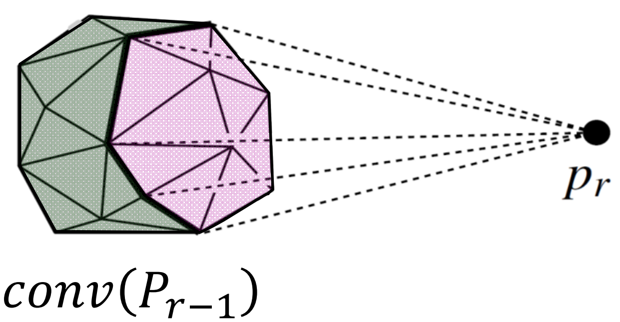
$conv(P_r)$

- Keep invisible facets
- Replace visible facets
  $\Rightarrow$ How to find all visible facets in time linear to their number?
  ($O(r-1)$ time is trivial but leads to an $O(n^2)$ algorithm.)

Conflict/Visibility Lists

Maintain conflict lists for each f on $conv(P_{r-1})$ and p_t for $t > r$:

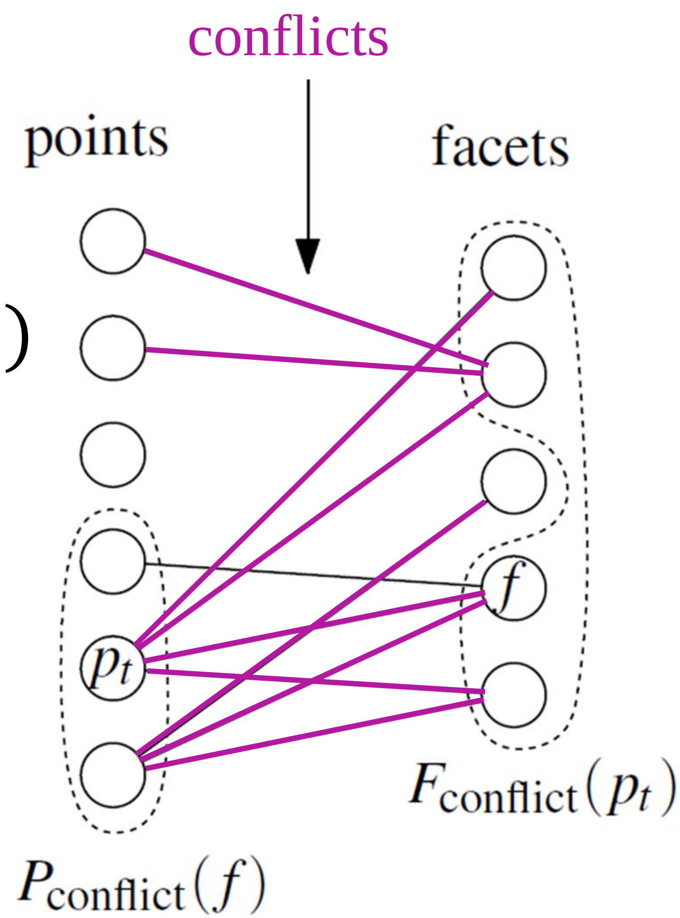
- $P_{conflict}(f) \subseteq \{p_r, \ldots, p_n\}$ consists of all points that can see f
- $F_{conflict}(p_t)$ consists of all facets of $conv(P_{r-1})$ visible from p_t
- $p \in P_{conflict}(f)$ is in conflict with f because p and f cannot be part of the same convex hull



Conflict Graph

Store all conflict lists in the conflict graph:

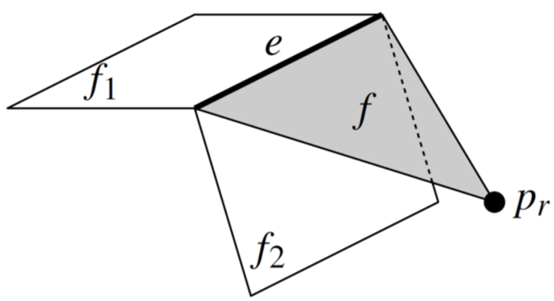
- Bipartite graph
- Node for every point in P that is not inserted yet
- Node for every facet of $conv(P_{r-1})$
- Arc between f and p if f is *in conflict with* (i.e., visible from) p

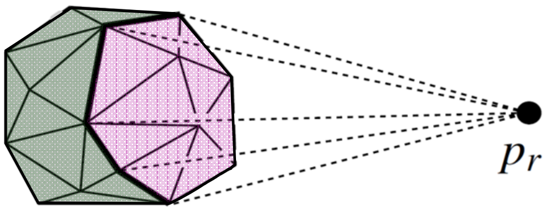


Maintaining the Conflict Graph

- Initialize conflict graph G for $conv(P_4)$ in linear time
- Update G after adding p_r:
  - Discard from G:
    - All neighbors of p_r in G. These are the facets visible from p_r.
    - p_r
  - Insert nodes in G for newly created facets (those facets which connect p_r to the horizon)
  - Find conflict lists for each newly created facet f:
    - A point p_t that sees f must also see e
    - But then p_t must have seen one of the faces f_1 or f_2 incident to e in $\mathrm{conv}(\mathrm{P}_{r-1})$

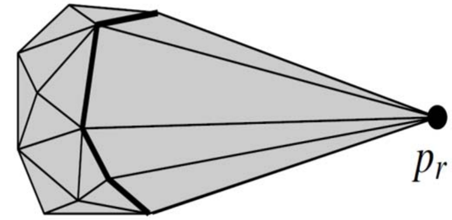
    $\Rightarrow$ Test all points in the conflict lists of f_1 and f_2





$conv(P_{r-1})$

Algorithm (part 1)



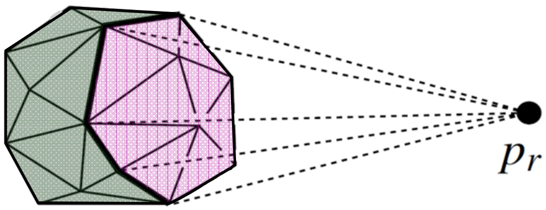
$conv(P_r)$

Algorithm CONVEXHULL(P)

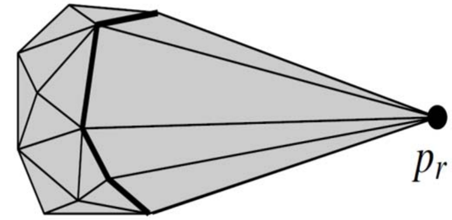
Input. A set P of n points in three-space.

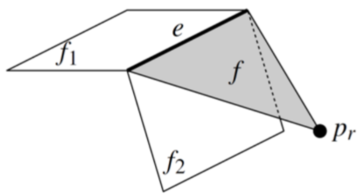
Output. The convex hull $\mathcal{CH}(P)$ of P.

1. Find four points p_1, p_2, p_3, p_4 in P that form a tetrahedron.
2. $\mathcal{C} \leftarrow \mathcal{CH}(\{p_1, p_2, p_3, p_4\})$
3. Compute a random permutation $p_5, p_6, \ldots, p_n$ of the remaining points.
4. Initialize the conflict graph $\mathcal{G}$ with all visible pairs (p_t, f), where f is a facet of $\mathcal{C}$ and $t > 4$.
5. **for** $r \leftarrow 5$ **to** n
6.   **do** (∗ Insert p_r into $\mathcal{C}$: ∗)
7.   **if** $F_{\text{conflict}}(p_r)$ is not empty (∗ that is, p_r lies outside $\mathcal{C}$ ∗)
8.   **then** Delete all facets in $F_{\text{conflict}}(p_r)$ from $\mathcal{C}$.
9.   Walk along the boundary of the visible region of p_r (which consists exactly of the facets in $F_{\text{conflict}}(p_r)$) and create a list $\mathcal{L}$ of horizon edges in order.
10.   **for** all $e \in \mathcal{L}$
11.   **do** Connect e to p_r by creating a triangular facet f.
12.   **if** f is coplanar with its neighbor facet f' along e
13.   **then** Merge f and f' into one facet, whose conflict list is the same as that of f'.



Algorithm (part 2)

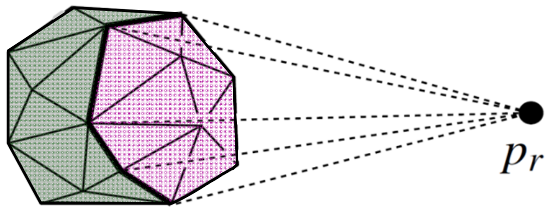




else (∗ Determine conflicts for f: ∗)
    Create a node for f in $\mathcal{G}$.
    Let f_1 and f_2 be the facets incident to e in the old convex hull.
    $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
    for all points $p \in P(e)$
        do If f is visible from p, add (p, f) to $\mathcal{G}$.

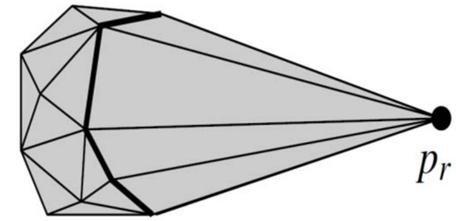
Delete the node corresponding to p_r and the nodes corresponding to the facets in $F_{\text{conflict}}(p_r)$ from $\mathcal{G}$, together with their incident arcs.

return $\mathcal{C}$



$conv(P_{r-1})$

Algorithm (part 1)



$conv(P_r)$

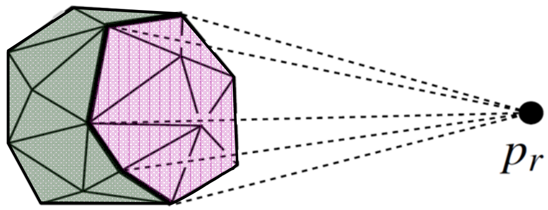
Algorithm CONVEXHULL(P)
Input. A set P of n points in three-space.
Output. The convex hull $\mathcal{CH}(P)$ of P.

$O(n)$

1. Find four points p_1, p_2, p_3, p_4 in P that form a tetrahedron.
2. $\mathcal{C} \leftarrow \mathcal{CH}(\{p_1, p_2, p_3, p_4\})$
3. Compute a random permutation $p_5, p_6, \dots, p_n$ of the remaining points.
4. Initialize the conflict graph $\mathcal{G}$ with all visible pairs (p_t, f), where f is a facet of $\mathcal{C}$ and $t > 4$.
5. **for** $r \leftarrow 5$ **to** n
6. **do** (* Insert p_r into $\mathcal{C}$: *)
7. **if** $F_{\text{conflict}}(p_r)$ is not empty (* that is, p_r lies outside $\mathcal{C}$ *)
8. **then** Delete all facets in $F_{\text{conflict}}(p_r)$ from $\mathcal{C}$.
9. Walk along the boundary of the visible region of p_r (which consists exactly of the facets in $F_{\text{conflict}}(p_r)$) and create a list $\mathcal{L}$ of horizon edges in order.
10. **for** all $e \in \mathcal{L}$
11. **do** Connect e to p_r by creating a triangular facet f.
12. **if** f is coplanar with its neighbor facet f' along e
13. **then** Merge f and f' into one facet, whose conflict list is the same as that of f'.

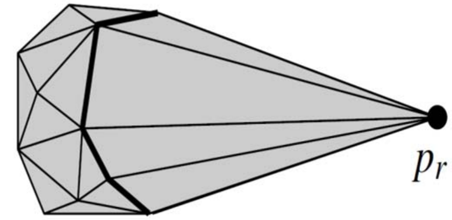
$O(|F_{conflict}(p_r)|)$

Face can only be deleted if it has been created.
→ Delete at most once



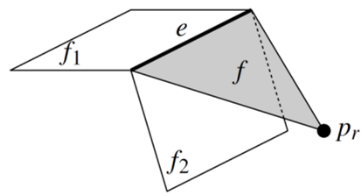
$conv(P_{r-1})$

Algorithm (part 2)



$conv(P_r)$

else (∗ Determine conflicts for f: ∗)
  Create a node for f in $\mathcal{G}$.
  Let f_1 and f_2 be the facets incident to e in the old convex hull.



$O(\sum_{e \in \mathcal{L}} |P(e)|)$
Charge to node and arc creation

  $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
  for all points $p \in P(e)$
    do If f is visible from p, add (p, f) to $\mathcal{G}$.

Delete the node corresponding to p_r and the nodes corresponding to the facets in $F_{\text{conflict}}(p_r)$ from $\mathcal{G}$, together with their incident arcs.

return $\mathcal{C}$

Need to bound:

- $O(n)$ • $E(\sum_{r=5}^{n} |F_{conflict}(p_r)|)$
- $O(n \log n)$ proof in book • $E(\sum_e |P(e)|)$ where the summation is over all horizon edges that ever appear during the algorithm

Backwards Analysis

Lemma: The expected number of facets created by the algorithm, $E(\sum_{r=1}^{n} |F_{conflict}(p_r)|)$, is at most $6n-20$.

Proof: $|F_{conflict}(p_r)|$

= # facets connecting p_r to its horizon on $conv(P_{r-1})$
(these are the newly created facets)

= # facets that disappear when removing p_r from $conv(P_r)$

=: $\deg(p_r, conv(P_r))$ *degree* of p_r in $conv(P_r)$

$\Rightarrow E(\deg(p_r, conv(P_r)) = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, conv(P_r))$

$\leq \frac{1}{r-4} [(\sum_{i=1}^{r} \deg(p_i, conv(P_r))) - 12]$ Total degree of $p_1, \ldots, p_4$ is at least 12

$\leq \frac{1}{r-4} [2(3r-6) - 12] = \frac{6r-12-12}{r-4} = 6$

Thus $4 + E\left(\sum_{r=5}^{n} |F_{conflict}(p_r)|\right) \leq 4 + 6(n-4) = 6n - 20$ □

Convex Hull Runtime

- **Theorem:** The convex hull of a set of n points in $\mathbb{R}^3$ can be computed in randomized expected $O(n \log n)$ time.
- **Theorem (higher dimensions):** The complexity of the convex hull of n points in $\mathbb{R}^d$ is $\Theta(n^{\lfloor d/2 \rfloor})$.
  It can be computed in randomized expected $O(n^{\lfloor d/2 \rfloor} + n \log n)$ time.