Computational Geometry
Winter semester 2016/17

Convex Hulls in 3D
Lecture #9
(Chapter 11)
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, 

Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. #edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
## Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
</tbody>
</table>

[Diagram showing the complexity of the convex hull in different dimensions]
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim $d$</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
# Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. \#edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td><strong>Your task!</strong></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1$</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>$2$</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>$3$</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^{\lfloor d/2 \rfloor})$</td>
</tr>
</tbody>
</table>
# Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>$w$-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^{\lfloor d/2 \rfloor})$</td>
</tr>
</tbody>
</table>

**Upper Bound Theorem**
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^{\lfloor d/2 \rfloor})$</td>
</tr>
</tbody>
</table>

Upper Bound Theorem

Construction?
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. \#edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^{\lfloor d/2 \rfloor})$</td>
</tr>
</tbody>
</table>

Upper Bound Theorem

Construction

randomized-incremental!
Visibility
Face $f$ is *visible* from $p$ but not from $q$. 
Visibility

Face $f$ is *visible* from $p$ but not from $q$. 
Visibility

Face $f$ is *visible* from $p$ but not from $q$. 
Visibility

Face $f$ is visible from $p$ but not from $q$. 
Visibility

Face $f$ is *visible* from $p$ but not from $q$. 

$h_f$
Visibility

Face $f$ is *visible* from $p$ but not from $q$. 
Visibility

Face $f$ is visible from $p$ but not from $q$.

Define conflict graph $G$: points (visibility) conflicts facets

points

facets

conflicts (visibility)

r

f
Visibility

Face \( f \) is visible from \( p \) but not from \( q \).

Define conflict graph \( G \):
Visibility

Face $f$ is *visible* from $p$ but not from $q$.

Define conflict graph $G$:
Rand3dConvexHull($P \subset \mathbb{R}^3$)
Rand3dConvexHull(\( P \subset \mathbb{R}^3 \))

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$
Rand3dConvexHull\((P \subset \mathbb{R}^3)\)

pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)

\(C \leftarrow \text{CH}(P')\)

compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$:

$(p, f)$ edge $\iff f$ visible from $p$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$:

- $(p, f)$ edge $\iff f$ visible from $p$

$P(e) \leftarrow P\text{-conflict}(f_1) \cup P\text{-conflict}(f_2)$

foreach $p \in P(e)$ do

if $f$ is visible from $p$ then

add edge $(p, f)$ to $G$

end if

end foreach

delete $\{p_r\} \cup F\text{-conflict}(p_r)$ from $G$

return $C \setminus \{p_r \in C\}$
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

- pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)
- \( C \leftarrow \text{CH}(P') \)
- compute rand. perm. \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)

initialize conflict graph \( G \)

for \( r = 5 \) to \( n \) do

\[
\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ \ p_r \notin C \ \}
\]

return \( C \)
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$
$C \leftarrow \text{CH}(P')$
compute rand. perm. $(p_5, \ldots, p_n)$ of $P \backslash P'$
initialize conflict graph $G$
for $r = 5$ to $n$ do
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then  
    \begin{align*}
    \{ \ p_r \not\in C \ 
    \}
  \end{align*}
return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)
-
pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$
-
compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$
-
initialize conflict graph $G$
-
for $r = 5$ to $n$ do
-
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \quad \{ $p_r \not\in C$ \}
-
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
-
return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. ($p_5, \ldots, p_n$) of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

end if

end for

return $C$
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)

\( C \leftarrow \text{CH}(P') \)

compute rand. perm. \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)

initialize conflict graph \( G \)

\[ \text{for } r = 5 \text{ to } n \text{ do} \]

\[ \quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \not\in C \} \]

\[ \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \]

\[ \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \]

return \( C \)
Rand3dConvexHull($P \subseteq \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

    if $F_{\text{conflict}}(p_r) \neq \emptyset$ then

        delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

        $\mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r$

return $C$
Rand3dConvexHull($P \subseteq \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

\begin{align*}
\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \not\in C \} \\
delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
\end{align*}

$L \leftarrow \text{list of horizon edges visible from } p_r$

foreach $e \in L$ do

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

    if $F_{\text{conflict}}(p_r) \neq \emptyset$ then

        delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

    foreach $e \in \mathcal{L}$ do

        $f \leftarrow C.\text{create facet}(e, p_r)$; create vtx for $f$ in $G$

return $C$
**Rand3dConvexHull**($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

**for** $r = 5$ **to** $n$ **do**

```
if $F_{\text{conflict}}(p_r) \neq \emptyset$ then { \{ $p_r \not\in C$ \}}
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
```

$L \leftarrow$ list of horizon edges visible from $p_r$

**foreach** $e \in L$ **do**

```
    f \leftarrow C\.create\_facet(e, p_r); create vtx for $f$ in $G$
    (f_1, f_2) \leftarrow \text{previously\_incident}_C(e)$
    P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
```

**return** $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

    if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \not\in C$ \}

    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

    foreach $e \in \mathcal{L}$ do

        $f \leftarrow C\.create\_facet(e, p_r)$; create vtx for $f$ in $G$

        $(f_1, f_2) \leftarrow \text{previously\_incident}_C(e)$

        $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

return $C$
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)

\( C \leftarrow \text{CH}(P') \)

compute rand. perm. \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)

initialize conflict graph \( G \)

\textbf{for} \( r = 5 \) \textbf{to} \( n \) \textbf{do}

\hspace{1em} \textbf{if} \( F_{\text{conflict}}(p_r) \neq \emptyset \) \textbf{then} \quad \{ \ p_r \notin C \ \}

\hspace{2em} delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)

\hspace{2em} \( \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \)

\hspace{2em} \textbf{foreach} \( e \in \mathcal{L} \) \textbf{do}

\hspace{3em} \( f \leftarrow C.\text{create}\_\text{facet}(e, p_r); \text{create vtx for } f \text{ in } G \)

\hspace{3em} \( (f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e) \)

\hspace{3em} \( P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \)

\textbf{return} \( C \)
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
  
  $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

foreach $e \in \mathcal{L}$ do

  $f \leftarrow C.\text{create}\_\text{facet}(e, p_r)$; create vtx for $f$ in $G$

  $(f_1, f_2) \leftarrow \text{previously}_\text{incident}_C(e)$

  $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

return $C$
Rand3dBConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \not\in C$ \}

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow$ list of horizon edges visible from $p_r$

foreach $e \in L$ do

$f \leftarrow \text{C.createfacet}(e, p_r)$; create vtx for $f$ in $G$

$(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

\hspace{1em} if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \not\in C$ \}

\hspace{2em} delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

\hspace{1em} $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

\hspace{1em} foreach $e \in \mathcal{L}$ do

\hspace{2em} $f \leftarrow C.\text{create facet}(e, p_r)$; create vtx for $f$ in $G$

\hspace{2em} $(f_1, f_2) \leftarrow \text{previously incident}_C(e)$

\hspace{2em} $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

\hspace{2em} foreach $p \in P(e)$ do

\hspace{3em} if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

\begin{itemize}
  \item if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{$p_r \not\in C$\}

  delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

  $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

  foreach $e \in \mathcal{L}$ do

  \begin{itemize}
    \item $f \leftarrow C.\text{create\_facet}(e, p_r)$; create vtx for $f$ in $G$
    \item $(f_1, f_2) \leftarrow$ previously\_incident$_C(e)$
    \item $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

    foreach $p \in P(e)$ do

    \begin{itemize}
      \item if $f$ is visible from $p$ then add edge $(p, f)$ to $G$
    \end{itemize}
  \end{itemize}

\end{itemize}

return $C$
Rand3dConvexHull($P \subseteq \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then  

$\{ p_r \not\in C \}$

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow$ list of horizon edges visible from $p_r$

foreach $e \in L$ do

$f \leftarrow C.\text{create\_facet}(e, p_r)$; create vtx for $f$ in $G$

$(f_1, f_2) \leftarrow \text{previously\_incident}_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

foreach $p \in P(e)$ do

if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

return $C$
Rand3dConvexHull\((P \subset \mathbb{R}^3)\)
pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)
\(C \leftarrow \text{CH}(P')\)
compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)
initialize conflict graph \(G\)
\(\text{for } r = 5 \text{ to } n \text{ do}\)
  \(\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \notin C \}\)
  delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)
  \(\mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r\)
  \(\text{foreach } e \in \mathcal{L} \text{ do}\)
    \(f \leftarrow C.\text{create\_facet}(e, p_r); \text{create vtx for } f \text{ in } G\)
    \((f_1, f_2) \leftarrow \text{previously\_incident}_C(e)\)
    \(P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)\)
    \(\text{foreach } p \in P(e) \text{ do}\)
      \(\text{if } f \text{ is visible from } p \text{ then add edge } (p, f) \text{ to } G\)
return \(C\)
Rand3dConvexHull\((P \subset \mathbb{R}^3)\)

pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)

\(C \leftarrow \text{CH}(P')\)

compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)

initialize conflict graph \(G\)

for \(r = 5\) to \(n\) do

  if \(F_{\text{conflict}}(p_r) \neq \emptyset\) then \(\{ p_r \not\in C \}\)

    delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)

    \(\mathcal{L} \leftarrow \) list of horizon edges visible from \(p_r\)

    foreach \(e \in \mathcal{L}\) do

      \(f \leftarrow C.\text{create\_facet}(e, p_r)\); create vtx for \(f\) in \(G\)

      \((f_1, f_2) \leftarrow \text{previously\_incident}_C(e)\)

      \(P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)\)

    foreach \(p \in P(e)\) do

      if \(f\) is visible from \(p\) then add edge \((p, f)\) to \(G\)

return \(C\)
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

\[
\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \not\in C \}
\]

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow \text{list of horizon edges visible from } p_r$

foreach $e \in L$ do

\[
f \leftarrow C.\text{create facet}\(e, p_r\); \text{create vtx for } f \text{ in } G
\]

\[
(f_1, f_2) \leftarrow \text{previously_incident}_C(e)
\]

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

foreach $p \in P(e)$ do

\[
\text{if } f \text{ is visible from } p \text{ then add edge } (p, f) \text{ to } G
\]

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then  

    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

    foreach $e \in \mathcal{L}$ do

      $f \leftarrow C.\text{create_facet}(e, p_r)$; create vtx for $f$ in $G$

      $(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$

      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

      foreach $p \in P(e)$ do

        if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

      delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

    if $F_{\text{conflict}}(p_r) \neq \emptyset$ then

        delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

        $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

        foreach $e \in \mathcal{L}$ do

            $f \leftarrow C\text{-create}\_\text{facet}(e, p_r)$; create vtx for $f$ in $G$

            $(f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e)$

            $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

            foreach $p \in P(e)$ do

                if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

            delete vtc $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$

return $C$

Worst-case running time =
Rand3dConvexHull\((P \subset \mathbb{R}^3)\)
pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)
\(C \leftarrow \text{CH}(P')\)
compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)
initialize conflict graph \(G\)
\textbf{for} \(r = 5\) \textbf{to} \(n\) \textbf{do}
\hspace{1em} \textbf{if} \(F_{\text{conflict}}(p_r) \neq \emptyset\) \textbf{then} \{ \(p_r \not\in C\) \}
\hspace{2em} delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)
\hspace{1em} \(\mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r\)
\hspace{1em} \textbf{foreach} \(e \in \mathcal{L}\) \textbf{do}
\hspace{2em} \(f \leftarrow C.\text{create} \_\text{facet}(e, p_r)\); create vtx for \(f\) in \(G\)
\hspace{2em} \((f_1, f_2) \leftarrow \text{previously} \_\text{incident}_C(e)\)
\hspace{2em} \(P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)\)
\hspace{2em} \textbf{foreach} \(p \in P(e)\) \textbf{do}
\hspace{3em} \textbf{if} \(f\) is visible from \(p\) \textbf{then} add edge \((p, f)\) to \(G\)
\hspace{2em} delete vtc \(\{p_r\} \cup F_{\text{conflict}}(p_r)\) from \(G\)
\textbf{return} \(C\)

Worst-case running time =
Rand3dConvexHull \((P \subset \mathbb{R}^3)\)

pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)

\(C \leftarrow \text{CH}(P')\)

compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)

initialize conflict graph \(G\)

for \(r = 5\) to \(n\) do

\[
\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \quad \{ \ p_r \notin C \ \}
\]

delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)

\(\mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r\)

foreach \(e \in \mathcal{L}\) do

\[
f \leftarrow C.\text{create\_facet}(e, p_r); \text{ create vtx for } f \text{ in } G
\]

\((f_1, f_2) \leftarrow \text{previously\_incident}_C(e)\)

\(P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)\)

foreach \(p \in P(e)\) do

\[
\text{if } f \text{ is visible from } p \text{ then add edge } (p, f) \text{ to } G
\]

delete vtc \(\{p_r\} \cup F_{\text{conflict}}(p_r)\) from \(G\)

return \(C\)

Worst-case running time = \(O(n^3)\)
Analysis

Idea: Bound expected *structural change*
Analysis

**Idea:** Bound expected *structural change*, that is, the total \#facets created by the algorithm.
Analysis

**Idea:** Bound expected *structural change*, that is, the total \#facets created by the algorithm.

**Lemma.** The expected \#facets created is at most $6n - 20$. 
Analysis

Idea: Bound expected *structural change*, that is, the total \( \# \) facets created by the algorithm.

Lemma. The expected \( \# \) facets created is at most \( 6n - 20 \).

Proof. \( E[\# \text{ facets created}] = \)
Analysis

Idea:  Bound expected *structural change*,
that is, the total \#facets created by the algorithm.

Lemma.  The expected \#facets created is at most $6n - 20$.

Proof.  
\[
E[\#\text{facets created}] = \\
= 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]
\]
Analysis

**Idea:** Bound expected *structural change*, that is, the total \#facets created by the algorithm.

**Lemma.** The expected \#facets created is at most $6n - 20$.

**Proof.**

$$E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in CH}(P_r)]$$
Analysis

**Idea:** Bound expected *structural change*, that is, the total #facets created by the algorithm.

**Lemma.** The expected #facets created is at most $6n - 20$.

**Proof.**

$$E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$$
Analysis

Idea: Bound expected *structural change*, that is, the total \#facets created by the algorithm.

Lemma. The expected \#facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$

$= 4 + \sum_{r=5}^{n} E[\#\text{edges}]$

$= 4 + \sum_{r=5}^{n} \text{deg}(p_r, \text{CH}(P_r))$
Analysis

**Idea:** Bound expected *structural change*, that is, the total #facets created by the algorithm.

**Lemma.** The expected #facets created is at most $6n - 20$.

**Proof.**

\[
E[\text{#facets created}] = 4 + \sum_{r=5}^{n} E[\text{#facets incident to } p_r \text{ in } CH(P_r)]
\]

For $r > 4$:

\[
E[\deg(p_r, CH(P_r))] =
\]
Analysis

Idea: Bound expected structural change, that is, the total #facets created by the algorithm.

Lemma. The expected #facets created is at most $6n - 20$.

Proof. 

$$E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$$

For $r > 4$:

$$E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r))$$
Analysis

Idea: Bound expected *structural change*, that is, the total \#facets created by the algorithm.

Lemma. The expected \#facets created is at most $6n - 20$.

Proof. 

\[
E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]
\]

For $r > 4$:

\[
E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r)) \leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \deg(p_i)) - 12 \right]
\]
Analysis

**Idea:** Bound expected *structural change*, that is, the total #facets created by the algorithm.

**Lemma.** The expected #facets created is at most $6n - 20$.

**Proof.**

$$E[\# \text{facets created}] = 4 + \sum_{r=5}^{n} E[\# \text{facets incident to } p_r \text{ in } CH(P_r)]$$

For $r > 4$:

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

$$\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \deg(p_i)) - 12 \right]$$
Analysis

Idea: Bound expected structural change, that is, the total #facets created by the algorithm.

Lemma. The expected #facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] =
\begin{align*}
&= 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } CH(P_r)] \\
&\text{For } r > 4:
E[\text{deg}(p_r, CH(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \text{deg}(p_i, CH(P_r)) \\
&\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \text{deg}(p_i)) - 12 \right] \\
&\leq 2 \cdot \# \text{ edges of } CH(P_r)
\end{align*}$
Analysis

Idea: Bound expected \textit{structural change}, that is, the total \# facets created by the algorithm.

Lemma. The expected \# facets created is at most $6n - 20$.

Proof. 
\[ E[\# \text{facets created}] = 4 + \sum_{r=5}^{n} E[\# \text{facets incident to } p_r \text{ in CH}(P_r)] \]

For $r > 4$:
\[ E[\text{deg}(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \text{deg}(p_i, \text{CH}(P_r)) \]
\[ \leq \frac{1}{r-4} \left[ \left( \sum_{i=1}^{r} \text{deg}(p_i) \right) - 12 \right] \]
\[ \leq \frac{1}{r-4} \left[ 2 \cdot \# \text{edges of CH}(P_r) \right] \]
\[ \leq \frac{1}{r-4} \left[ 2 \cdot (3r - 6) - 12 \right] \]
Analysis

**Idea:** Bound expected *structural change*, that is, the total #facets created by the algorithm.

**Lemma.** The expected #facets created is at most $6n - 20$.

**Proof.** \[
E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]
\]

For $r > 4$:
\[
E[\text{deg}(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \text{deg}(p_i, \text{CH}(P_r)) \leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \text{deg}(p_i)) - 12 \right]
\]
\[
\leq \frac{1}{r-4} \left[ 2 \cdot \text{# edges of } \text{CH}(P_r) \right] \leq 6
\]
Analysis

**Idea:** Bound expected *structural change*, that is, the total #facets created by the algorithm.

**Lemma.** The expected #facets created is at most $6n - 20$.

**Proof.**

$E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)] \leq \#\text{edges}$

For $r > 4$:

$E[\text{deg}(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \text{deg}(p_i, \text{CH}(P_r))$

$\leq \frac{1}{r-4} \left[\left(\sum_{i=1}^{r} \text{deg}(p_i)\right) - 12\right]$

$\leq \frac{1}{r-4} \left[2 \cdot (3r - 6) - 12\right] \leq 6$
Analysis

Idea: Bound expected *structural change*, that is, the total #facets created by the algorithm.

Lemma. The expected #facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)] \leq 6n - 20$

For $r > 4$:

$E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r))$

$\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \deg(p_i)) - 12 \right]$

$\leq \frac{1}{r-4} \left[ 2 \cdot \text{edges of } \text{CH}(P_r) \right]$

$\leq \frac{1}{r-4} \left[ 2 \cdot (3r - 6) - 12 \right] \leq 6$
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull($P \subset \mathbb{R}^3$)
    pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
    $C \leftarrow \text{CH}(P')$
    compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
    initialize conflict graph $G$
    for $r = 5$ to $n$ do
        if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
            delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
            $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
            foreach $e \in \mathcal{L}$ do
                $f \leftarrow C.\text{create_facet}(e, p_r)$; create vtx for $f$ in $G$
                $(f_1, f_2) \leftarrow \text{previously Incident}_C(e)$
                $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
                foreach $p \in P(e)$ do
                    if $f$ visible from $p$ then add edge $(p, f)$ to $G$
            delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
    return $C$
```
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull(P \subseteq \mathbb{R}^3)
{
    pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
    $C \leftarrow \text{CH}(P')$
    compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
    initialize conflict graph $G$
    for $r = 5$ to $n$ do
        if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
            delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
            $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
            foreach $e \in \mathcal{L}$ do
                $f \leftarrow C$.create_facet($e, p_r$); create vtx for $f$ in $G$
                $(f_1, f_2) \leftarrow \text{previously}_\text{incident}_C(e)$
                $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
                foreach $p \in P(e)$ do
                    if $f$ visible from $p$ then add edge $(p, f)$ to $G$
            delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
        }
    return $C$
}
```
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```plaintext
Rand3dConvexHull(\mathcal{P} \subset \mathbb{R}^3)
\begin{aligned}
&\text{pick set } \mathcal{P}' = \{p_1, \ldots, p_4\} \subseteq \mathcal{P} \text{ of 4 non-coplanar pts} \\
&C \leftarrow \text{CH}(\mathcal{P}') \\
&\text{compute a random permutation } (p_5, \ldots, p_n) \text{ of } \mathcal{P} \setminus \mathcal{P}' \\
&\text{initialize conflict graph } G \\
\text{for } r = 5 \text{ to } n \text{ do} \\
&\quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
&\quad\quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
&\quad\quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
&\quad \text{foreach } e \in \mathcal{L} \text{ do} \\
&\quad\quad f \leftarrow C.\text{create facet}(e, p_r); \text{ create vtx for } f \text{ in } G \\
&\quad\quad (f_1, f_2) \leftarrow \text{previously\_incident}_C(e) \\
&\quad\quad \mathcal{P}(e) \leftarrow \mathcal{P}_{\text{conflict}}(f_1) \cup \mathcal{P}_{\text{conflict}}(f_2) \\
&\quad\quad \text{foreach } p \in \mathcal{P}(e) \text{ do} \\
&\quad\quad\quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
&\quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\end{aligned}
```

$O(n)$ time
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop)

```latex
\begin{align*}
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3) & \quad \text{pick set } P' = \{p_1, \ldots, p_4\} \subseteq P \text{ of 4 non-coplanar pts} \\
& \quad C \leftarrow \text{CH}(P') \\
& \quad \text{compute a random permutation } (p_5, \ldots, p_n) \text{ of } P \setminus P' \\
& \quad \text{initialize conflict graph } G \\
& \quad \text{for } r = 5 \text{ to } n \text{ do} \\
& \quad \quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
& \quad \quad \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
& \quad \quad \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
& \quad \quad \quad \text{foreach } e \in \mathcal{L} \text{ do} \\
& \quad \quad \quad \quad f \leftarrow C.\text{create}\_\text{facet}(e, p_r); \text{ create vtx for } f \text{ in } G \\
& \quad \quad \quad \quad (f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e) \\
& \quad \quad \quad \quad P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
& \quad \quad \quad \text{foreach } p \in P(e) \text{ do} \\
& \quad \quad \quad \quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
& \quad \quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\end{align*}
```

$O(n)$ time

Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(n)$ time.

```
Rand3dConvexHull(P ⊂ \mathbb{R}^3)
    pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
    $C \leftarrow \text{CH}(P')$
    compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
    initialize conflict graph $G$
    for $r = 5$ to $n$ do
        if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
            delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
            $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
            foreach $e \in \mathcal{L}$ do
                $f \leftarrow C.\text{create facet}(e, p_r)$; create vtx for $f$ in $G$
                $(f_1, f_2) \leftarrow \text{previously incident}_C(e)$
                $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
                foreach $p \in P(e)$ do
                    if $f$ visible from $p$ then add edge $(p, f)$ to $G$
            delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
        return $C$
```
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r)$
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total:

```
Rand3dConvexHull(\mathbb{P} \subset \mathbb{R}^3)
pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
$C \leftarrow \text{CH}(P')$
compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
initialize conflict graph $G$

for $r = 5$ to $n$ do
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
    foreach $e \in \mathcal{L}$ do
      $f \leftarrow C.\text{create facet}(e, p_r)$; create vtx for $f$ in $G$
      $(f_1, f_2) \leftarrow \text{previously incident}_C(e)$
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ do
        if $f$ visible from $p$ then add edge $(p, f)$ to $G$
      delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
  return $C$
```
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull(\ P \subset \mathbb{R}^3) does
pick set \ P' = \{p_1, \ldots, p_4\} \subseteq P of 4 non-coplanar pts
C \leftarrow CH(P')
compute a random permutation \ (p_5, \ldots, p_n) of \ P \setminus P'
initialize conflict graph \ G
for \ r = 5 \ to \ n \ do
    if \ F_{\text{conflict}}(p_r) \neq \emptyset \ then
        delete all facets in \ F_{\text{conflict}}(p_r) from \ C
    \mathcal{L} \leftarrow \text{list of horizon edges visible from} \ p_r
    \text{foreach} \ e \in \mathcal{L} \ do
        f \leftarrow C.\text{create_facet}(e, p_r); \text{create vtx for} \ f \ in \ G
        (f_1, f_2) \leftarrow \text{previously_incident}_C(e)
        P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)
        \text{foreach} \ p \in P(e) \ do
            if \ f \ \text{visible from} \ p \ \text{then} \text{add edge} \ (p, f) \ \text{to} \ G
    delete vtc \ \{p_r\} \cup F_{\text{conflict}}(p_r) \ \text{from} \ G
return \ C
```

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding} \ p_r)$

This part of for-loop in total: $E[\#\text{facets deleted}] = \ldots$
Running Time

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

\[
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3) \\
\text{pick set } P' = \{p_1, \ldots, p_4\} \subseteq P \text{ of 4 non-coplanar pts} \\
C \leftarrow \text{CH}(P') \\
\text{compute a random permutation } (p_5, \ldots, p_n) \text{ of } P \setminus P' \\
\text{initialize conflict graph } G \\
\text{for } r = 5 \text{ to } n \text{ do} \\
\quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
\quad \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
\quad \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
\quad \text{foreach } e \in \mathcal{L} \text{ do} \\
\quad \quad f \leftarrow C.\text{create_facet}(e, p_r); \text{ create vtx for } f \text{ in } G \\
\quad \quad (f_1, f_2) \leftarrow \text{previously_incident}_C(e) \\
\quad \quad P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
\quad \quad \text{foreach } p \in P(e) \text{ do} \\
\quad \quad \quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
\quad \quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\text{return } C
\]

Stage \( r \) of for-loop (w/o outer foreach loop) takes time \( O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r) \)

This part of for-loop in total: \( E[\#\text{facets deleted}] = \)

\[
\leq E[\#\text{facets created}] =
\]
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull(P ⊂ \mathbb{R}^3)
{ pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts }
{ compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$ }
{ initialize conflict graph $G$ }
for $r = 5$ to $n$ do
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
    foreach $e \in \mathcal{L}$ do
      $f \leftarrow C.\text{create_facet}(e, p_r)$; create vtx for $f$ in $G$
      $(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ do
        if $f$ visible from $p$ then add edge $(p, f)$ to $G$
    delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
  return $C$
```

Stage $r$ of for-loop (w/o outer foreach loop)
takes time $O(\lvert F_{\text{conflict}}(p_r) \rvert) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total:

$E[\#\text{facets deleted}] = \leq E[\#\text{facets created}]$
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull(P \subset \mathbb{R}^3)
pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
$C \leftarrow \text{CH}(P')$
compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
initialize conflict graph $G$
for $r = 5$ to $n$ do
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
  $L \leftarrow$ list of horizon edges visible from $p_r$
  foreach $e \in L$ do
    $f \leftarrow \text{C.create facet}(e, p_r)$; create vtx for $f$ in $G$
    $(f_1, f_2) \leftarrow \text{previously incident}_C(e)$
    $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
    foreach $p \in P(e)$ do
      if $f$ visible from $p$ then add edge $(p, f)$ to $G$
  delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
return $C$
```

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total:

$E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n)$.

---

Lemma
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```latex
Rand3dConvexHull(P \subset \mathbb{R}^3)
\begin{enumerate}
\item pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
\item compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
\item initialize conflict graph $G$
\item for $r = 5$ to $n$ do
\begin{enumerate}
\item if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
\begin{enumerate}
\item delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
\item $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
\item foreach $e \in \mathcal{L}$ do
\begin{enumerate}
\item $f \leftarrow C.\text{create}\_\text{facet}(e, p_r)$; create vtx for $f$ in $G$
\item $(f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e)$
\item $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
\item foreach $p \in P(e)$ do
\begin{enumerate}
\item if $f$ visible from $p$ then add edge $(p, f)$ to $G$
\end{enumerate}
\end{enumerate}
\end{enumerate}
\end{enumerate}
\item delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total: $E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n)$.

Outer foreach-loop:
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull(P ⊂ \mathbb{R}^3)
{ pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
    $C \leftarrow \text{CH}(P')$
    compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
    initialize conflict graph $G$
    for $r = 5$ to $n$ do
        if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
            delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
            $\mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r$
            foreach $e \in \mathcal{L}$ do
                $f \leftarrow C.\text{create_facet}(e, p_r)$; create vtx for $f$ in $G$
                $(f_1, f_2) \leftarrow \text{previously_incident}_{C}(e)$
                $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
                foreach $p \in P(e)$ do
                    if $f$ visible from $p$ then add edge $(p, f)$ to $G$
            delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
        return $C$
}
```

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total: $E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n)$.  

**Lemma**

Outer foreach-loop: in stage $r$: $O(\sum_{e \in \mathcal{L}} |P(e)|)$
Running Time

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

```plaintext
Rand3dConvexHull(\( P \subset \mathbb{R}^3 \))
pick set \( P' = \{p_1, \ldots, p_4\} \subseteq P \) of 4 non-coplanar pts
\( C \leftarrow \text{CH}(P') \)
compute a random permutation \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)
initialize conflict graph \( G \)
for \( r = 5 \) to \( n \) do
  if \( \text{F\text{conflict}}(p_r) \neq \emptyset \) then
    delete all facets in \( \text{F\text{conflict}}(p_r) \) from \( C \)
    \( \mathcal{L} \leftarrow \) list of horizon edges visible from \( p_r \)
    foreach \( e \in \mathcal{L} \) do
      \( f \leftarrow C . \text{create_facet}(e, p_r) \); create vtx for \( f \) in \( G \)
      \( (f_1, f_2) \leftarrow \text{previously incident}_C(e) \)
      \( P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \)
      foreach \( p \in P(e) \) do
        if \( f \) visible from \( p \) then add edge \( (p, f) \) to \( G \)
    delete vtc \( \{p_r\} \cup \text{F\text{conflict}}(p_r) \) from \( G \)
return \( C \)
```

Stage \( r \) of for-loop (w/o outer foreach loop) takes time \( O(|\text{F\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r) \)

This part of for-loop in total:
\[
E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n).
\]

Lemma

Outer foreach-loop:
- in stage \( r \): \( O(\sum_{e \in \mathcal{L}} |P(e)|) \)
- in total:
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

\[
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3)
\]

- pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
- compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
- initialize conflict graph $G$

for $r = 5$ to $n$ do

\[
\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then}
\]
- delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
- $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

foreach $e \in \mathcal{L}$ do

\[
\text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G
\]

delete vtc \{p_r\} $\cup$ $F_{\text{conflict}}(p_r)$ from $G$

return $C$

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total:

\[
E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n).
\]

Outer foreach-loop:

- in stage $r$: $O(\sum_{e \in \mathcal{L}} |P(e)|)$
- in total:

\[
O\left(\sum_{e \text{ on horizon at some moment}} |P(e)|\right)
\]
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull(P \subset \mathbb{R}^3)

pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
C ← CH($P'$)
compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
initialize conflict graph $G$
for $r = 5$ to $n$ do
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
    $L$ ← list of horizon edges visible from $p_r$
    foreach $e \in L$ do
      $f ← C.\text{create\_facet}(e, p_r)$; create vtx for $f$ in $G$
      $(f_1, f_2) ← \text{previously\_incident}_C(e)$
      $P(e) ← P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ do
        if $f$ visible from $p$ then add edge $(p, f)$ to $G$
    delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
return $C$
```

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r)$

This part of for-loop in total:

$E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n)$.  

Lemma

Outer foreach-loop:

- in stage $r$: $O(\sum_{e \in L} |P(e)|)$
- in total:

$$O \left( \sum_{e \text{ on horizon at some moment}} |P(e)| \right) = O(n \log n)$$
Running Time

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

Stage \( r \) of for-loop (w/o outer foreach loop) takes time \( O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r) \)

This part of for-loop in total: \( E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n). \)

Outer foreach-loop:
- in stage \( r \): \( O(\sum_{e \in \mathcal{L}} |P(e)|) \)
- in total:

\[
O \left( \sum_{\text{e on horizon at some moment}} |P(e)| \right) = O(n \log n)
\]

using configuration spaces, Section 9.5 [De Berg et al.]
Running Time

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

```plaintext
Rand3dConvexHull(P ⊂ \mathbb{R}^3) 
pick set \( P' = \{p_1, \ldots, p_4\} \subseteq P \) of 4 non-coplanar pts
C ← CH(\( P' \))
compute a random permutation \((p_5, \ldots, p_n)\) of \( P \setminus P' \)
initialize conflict graph \( G \)
for \( r = 5 \) to \( n \) do
  if \( F_{\text{conflict}}(p_r) \neq \emptyset \) then
    delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)
  \( \mathcal{L} \) ← list of horizon edges visible from \( p_r \)
  foreach \( e \in \mathcal{L} \) do
    \( f \leftarrow C.\text{create-facet}(e, p_r) \); create vtx for \( f \) in \( G \)
    \( (f_1, f_2) \leftarrow \text{previously-incident}_C(e) \)
    \( P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \)
    foreach \( p \in P(e) \) do
      if \( f \) visible from \( p \) then add edge \((p, f)\) to \( G \)
    delete vtx \{\( p_r \}\} ∪ \( F_{\text{conflict}}(p_r) \) from \( G \)
return \( C \)
```

Stage \( r \) of for-loop (w/o outer foreach loop)
takes time \( O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r) \)

This part of for-loop in total:
\( E[\#\text{facets deleted}] = \sum_{e \in \mathcal{L}} |P(e)| \)

Outer foreach-loop:
- in stage \( r \): \( O(\sum_{e \in \mathcal{L}} |P(e)|) \)
- in total:

\[
O \left( \sum_{e \text{ on horizon at some moment}} |P(e)| \right) = O(n \log n)
\]

using configuration spaces, Section 9.5 [De Berg et al.]
Theorem: The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.
Running times – expected vs. worst case

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

**Exercise:** Give a simple deterministic algorithm that computes the convex hull in $O(n^2)$ (worst-case) time.
Convex Hulls and Half-Space Intersections
Convex Hulls and Half-Space Intersections
Plane
Convex Hulls and Half-Space Intersections

Plane

Define duality $\star$ between pts and (non-vertical) lines:
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$,
Define duality \( \ast \) between pts and (non-vertical) lines:

For \( p = (p_x, p_y) \),
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$. 

![Diagram showing point $p$ and its primal representation](image-url)
Convex Hulls and Half-Space Intersections

Plane

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star$: $y = p_x x - p_y$. 
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$. 
Convex Hulls and Half-Space Intersections

Plane

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star$: $y = p_x x - p_y$.

For $\ell: y = mx + b$, 

\[\text{primal} \quad \text{dual} \]
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, 

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$. 

![Diagram showing primal and dual spaces](image_url)
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$. 
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$. 
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.  

Convex Hulls and Half-Space Intersections

Plane
Define duality $\ast$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\ast: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\ast$ to be the pt $q$ with $q^\ast = \ell$, that is, $\ell^\ast = (m, -b)$.

Observe:
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line. $\star$ is incidence-preserving:
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line. $\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star$: $y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.
$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$
$\star$ is order-preserving:
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.
$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$
$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Convex Hulls and Half-Space Intersections

Define duality \( \star \) between pts and (non-vertical) lines:

For \( p = (p_x, p_y) \), define the line \( p^*: y = p_x x - p_y \).

For \( \ell: y = mx + b \), define \( \ell^* \) to be the pt \( q \) with \( q^* = \ell \), that is, \( \ell^* = (m, -b) \).

Observe: Let \( p \in \mathbb{R}^2 \) and let \( \ell \) be a non-vertical line.
\( \star \) is incidence-preserving: \( p \in \ell \iff \ell^* \in p^* \)
\( \star \) is order-preserving: \( p \) above \( \ell \) \iff \( \ell^* \) above \( p^* \)
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star : y = p_x x - p_y$.

For $\ell : y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star$: $y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p \text{ above } \ell \iff \ell^\star \text{ above } p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Convex Hulls and Half-Space Intersections

Define duality \( \star \) between pts and (non-vertical) lines:

For \( p = (p_x, p_y) \), define the line \( p^* : y = p_x x - p_y \).

For \( \ell : y = mx + b \), define \( \ell^* \) to be the pt \( q \) with \( q^* = \ell \), that is, \( \ell^* = (m, -b) \).

Observe: Let \( p \in \mathbb{R}^2 \) and let \( \ell \) be a non-vertical line.

\( \star \) is incidence-preserving: \( p \in \ell \iff \ell^* \in p^* \)

\( \star \) is order-preserving: \( p \) above \( \ell \iff \ell^* \) above \( p^* \)
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^* : y = p_x x - p_y$.

For $\ell : y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$
$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

- $\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$
- $\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^* : y = p_x x - p_y$.

For $\ell : y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

- $\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$
- $\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^* : y = p_x x - p_y$.

For $\ell : y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.
- $\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$
- $\star$ is order-preserving: $\ p \text{ above } \ell \iff \ell^* \text{ above } p^*$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** upper convex hulls of pts $\leftrightarrow$ lower envelopes of lines
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:**
- upper convex hulls of pts $\leftrightarrow$ lower envelopes of lines
- can compute intersections of “lower/upper” half planes (spaces) via upper/lower convex hulls
Voronoi Diagrams Revisited

Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$. 
Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$. 
Voronoi Diagrams Revisited

Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$. 

$p = (p_x, p_y, 0)$
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$. 

$p = (p_x, p_y, 0)$
Voronoi Diagrams Revisited

Let \( U: z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
\begin{align*}
  p' &= (p_x, p_y, p_x^2 + p_y^2) \\
p &= (p_x, p_y, 0)
\end{align*}
\]
Voronoi Diagrams Revisited

Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

$p' = (p_x, p_y, p_x^2 + p_y^2)$

$p = (p_x, p_y, 0)$
Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$p' = (p_x, p_y, p_x^2 + p_y^2)$

$p = (p_x, p_y, 0)$
Let $U : z = x^2 + y^2$ be the *unit paraboloid* in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$p' = (p_x, p_y, p_x^2 + p_y^2)$

$p = (p_x, p_y, 0)$
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$p' = (p_x, p_y, p_x^2 + p_y^2)$

$p = (p_x, p_y, 0)$
Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
\begin{align*}
h(p) : z &= (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \\
\text{Note that } p' &\in h(p).
\end{align*}
\]

\[
p' = (p_x, p_y, p_x^2 + p_y^2)
\]

\[
p = (p_x, p_y, 0)
\]

\[
|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}
\]
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)
\]

Note that \( p' \in h(p) \).

\[
p' = (p_x, p_y, p_x^2 + p_y^2)
\]

\[
p = (p_x, p_y, 0)
\]

\[
|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}
\]
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)
\]

Note that \( p' \in h(p) \).
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
q(p) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}
\]

\[
q' = (p_x, p_y, p_x^2 + p_y^2)
\]

Note that \( p' \in h(p) \).

\[
h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)
\]
Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

\[ q' = (p_x, p_y, p_x^2 + p_y^2) \]

\[ q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2)) \]
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))
\]

\[
p = (p_x, p_y, 0)
\]

\[
|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}
\]

\[
h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)
\]

Note that \( p' \in h(p) \).
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[ h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \]

Note that \( p' \in h(p) \).

\[ q(p' = (p_x, p_y, p_x^2 + p_y^2) \]

\[ q(p) = (q_x, q_y, 2p_xq_x + 2p_yq_y - (p_x^2 + p_y^2)) \]

\[ |pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \]

\[ z_{q'} - z_{q(p)} = |pq|^2 \]
Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$h(p)$ and $U$ encode dist. betw. $p$ and any other pt in $z = 0$. 
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

Let $p = (p_x, p_y, 0)$ and $q = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))$

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$

$z_{q'} - z_{q(p)} = |pq|^2$

$h(p)$ and $U$ encode dist. betw. $p$ and any other pt in $z = 0$.

$h(p) \cap U = \{p'\}$
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

Note that $p' \in h(p)$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

$h(p) \cap U = \{p'\} \Rightarrow h(p)$ is tangent to $U$ (in $p'$)
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
\begin{align*}
\text{Let } & p = (p_x, p_y, 0) \\
\text{Note that } & p' \in h(p).
\end{align*}
\]

\[
\begin{align*}
h(q) = (q_x, q_y, 2p_xq_x + 2p_yq_y - (p_x^2 + p_y^2)) \\
\Rightarrow & h(p) \text{ and } U \text{ encode dist. betw. } p \text{ and any other pt in } z = 0. \\
\Rightarrow & h(p) \cap U = \{p'\} \Rightarrow h(p) \text{ is tangent to } U \text{ (in } p')
\end{align*}
\]
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$h(p) \cap U = \{p'\} \Rightarrow h(p)$ is tangent to $U$ (in $p'$)
Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))$

$\Rightarrow h(p)$ and $U$ encode dist. betw. $p$ and any other pt in $z = 0$.

$\Rightarrow h(p) \cap U = \{p'\} \Rightarrow h(p)$ is tangent to $U$ (in $p'$)
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{ h(p) \mid p \in P \}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$.
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$.
Theorem: Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$.
The Upper Envelope Strikes Back

**Theorem:** Let \( P \subset \mathbb{R}^2 \times \{0\} \) and \( \mathcal{H} = \{ h(p) \mid p \in P \} \).

Let \( \mathcal{E}(\mathcal{H}) \) be the upper envelope of \( \mathcal{H} \).

The projection of \( \mathcal{E}(\mathcal{H}) \) on \( z = 0 \) is...
**The Upper Envelope Strikes Back**

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$. 

\[ H = \{ h(p) \mid p \in P \} \]
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{ h(p) \mid p \in P \}$.

Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$.

The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

can compute $\text{Vor}(P)$ in $\mathbb{R}^2$ via upper envelope in $\mathbb{R}^3$
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

can compute $\text{Vor}(P)$ in $\mathbb{R}^2$ via upper envelope in $\mathbb{R}^3$

exercise 11.10
The Upper Envelope Strikes Back

**Theorem:** Let $P \subseteq \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

- can compute $\text{Vor}(P)$ in $\mathbb{R}^2$
- via upper envelope in $\mathbb{R}^3$
- exercise 11.10
- upper envelope in $\mathbb{R}^3$ is in one-to-one correspondence to lower convex hull of the pt set $\mathcal{H}^*$
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{ h(p) \mid p \in P \}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

- can compute $\text{Vor}(P)$ in $\mathbb{R}^2$
- via upper envelope in $\mathbb{R}^3$

exercise 11.10

upper envelope in $\mathbb{R}^3$ is in one-to-one correspondence to lower convex hull of the pt set $\mathcal{H}^*$
The Upper Envelope Strikes Back

**Theorem:** Let \( P \subset \mathbb{R}^2 \times \{0\} \) and \( \mathcal{H} = \{ h(p) \mid p \in P \} \).

Let \( \mathcal{E}(\mathcal{H}) \) be the upper envelope of \( \mathcal{H} \).

The projection of \( \mathcal{E}(\mathcal{H}) \) on \( z = 0 \) is \( \text{Vor}(P) \).

- Can compute \( \text{Vor}(P) \) in \( \mathbb{R}^2 \) via upper envelope in \( \mathbb{R}^3 \).
- Exercise 11.10

Upper envelope in \( \mathbb{R}^3 \) is in one-to-one correspondence to lower convex hull of the pt set \( \mathcal{H}^* \).

Use algorithm Rand3dConvexHull!