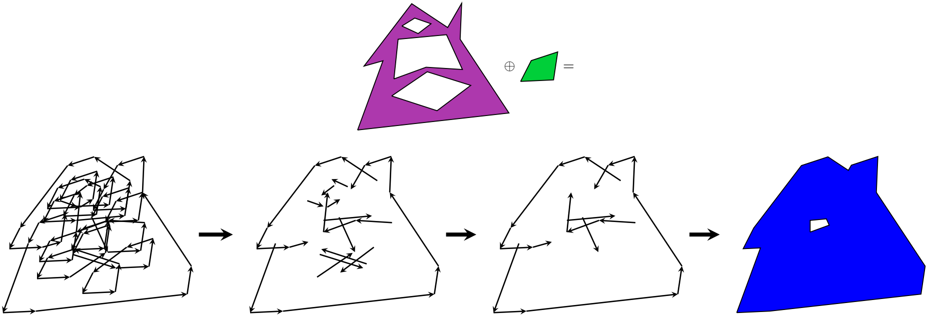


Exact Minkowski Sums of Polygons With Holes

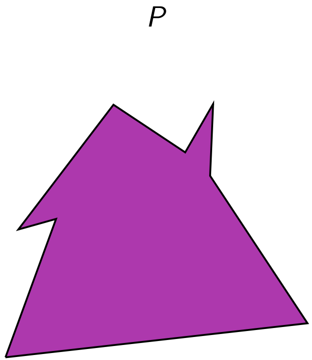
Alon Baram [1] Efi Fogel [1] Dan Halperin [1]
Michael Hemmer [2] Sebastian Morr [2]

[1] TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

[2] Technische Universität Braunschweig

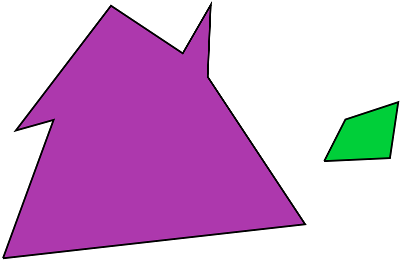
What's the Minkowski Sum?

What's the Minkowski Sum?



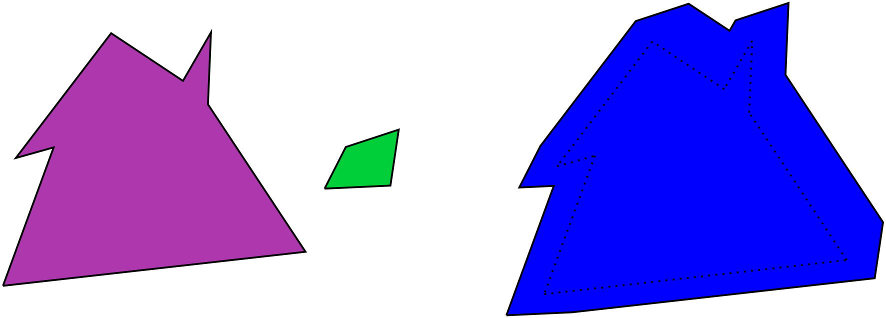
What's the Minkowski Sum?

$P \quad \oplus \quad Q$



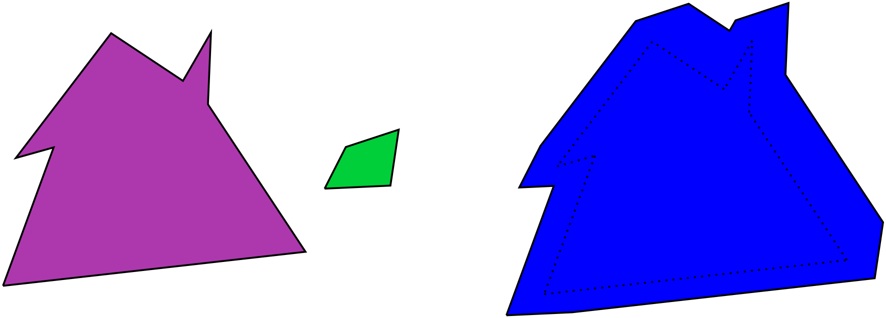
What's the Minkowski Sum?

$$P \oplus Q = \{p + q \mid p \in P, q \in Q\}$$



What's the Minkowski Sum?

$$P \quad \oplus \quad Q \quad = \quad \{p + q \mid p \in P, q \in Q\}$$

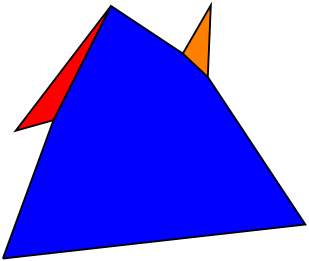


Applications: Motion planning, packing problems, CAD, ...

Decomposition approach

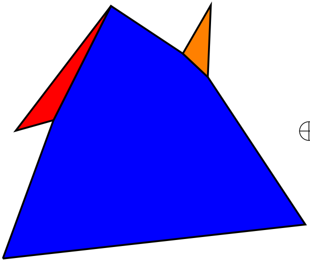
[Lozano-Pérez (1983)]

Decomposition approach

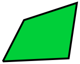


[Lozano-Pérez (1983)]

Decomposition approach



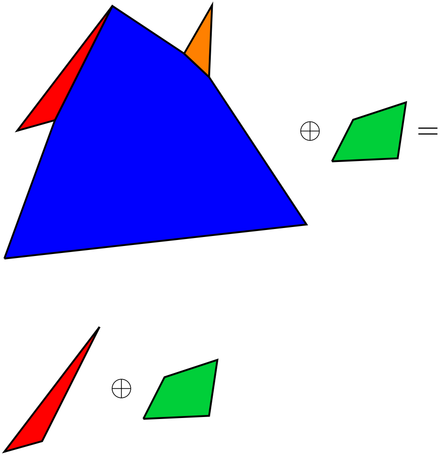
$\oplus$



$=$

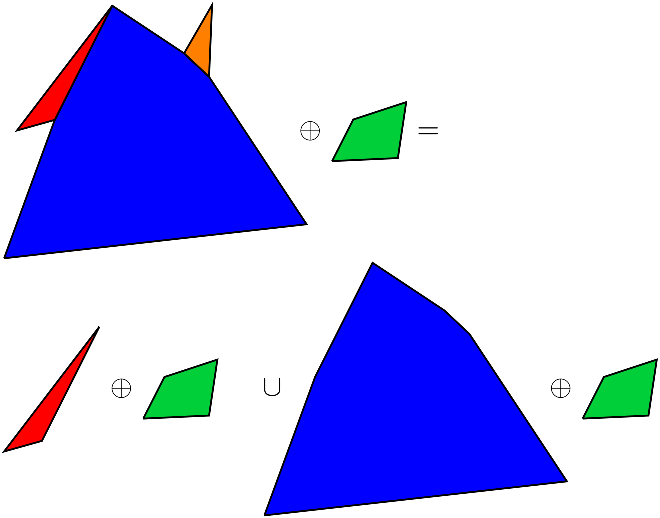
[Lozano-Pérez (1983)]

Decomposition approach



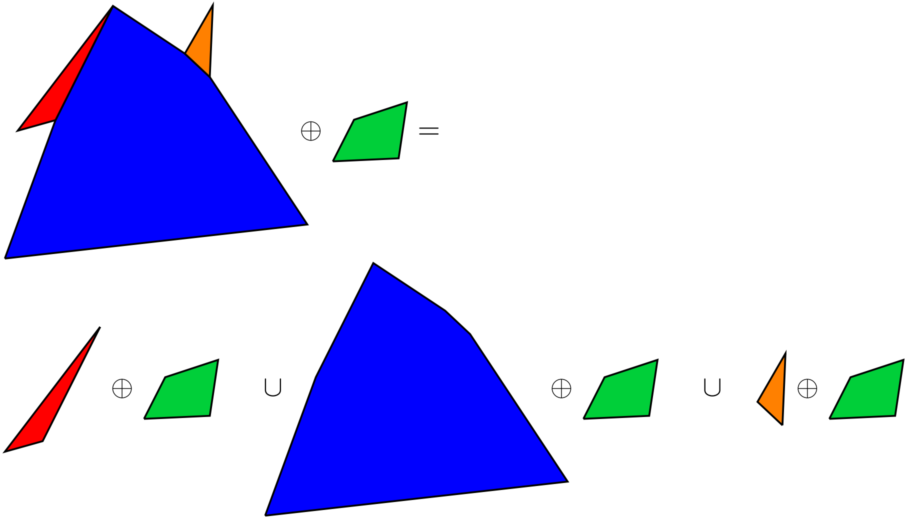
[Lozano-Pérez (1983)]

Decomposition approach



[Lozano-Pérez (1983)]

Decomposition approach



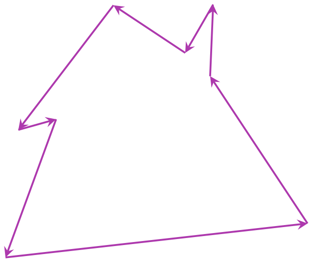
[Lozano-Pérez (1983)]

Convolution approach

[Guibas et al. (1983)]

Convolution approach

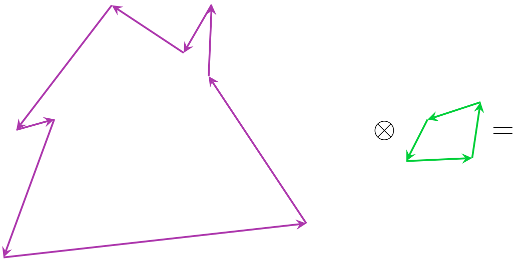
$$P = (p_0, p_1, \ldots, p_{n-1})$$



[Guibas et al. (1983)]

Convolution approach

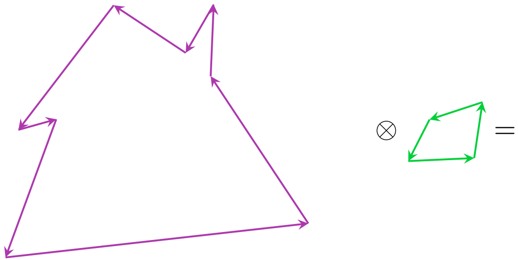
$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1})
\end{aligned}
$$



[Guibas et al. (1983)]

Convolution approach

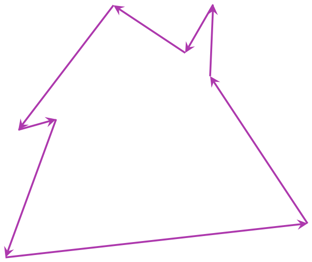
$$\begin{aligned} P &= (p_0, p_1, \ldots, p_{n-1}) \\ Q &= (q_0, q_1, \ldots, q_{m-1}) \\ P \otimes Q &= \{ \end{aligned}$$



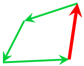
[Guibas et al. (1983)]

Convolution approach

$$\begin{aligned} P &= (p_0, p_1, \ldots, p_{n-1}) \\ Q &= (q_0, q_1, \ldots, q_{m-1}) \\ P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \end{aligned}$$



$\otimes$

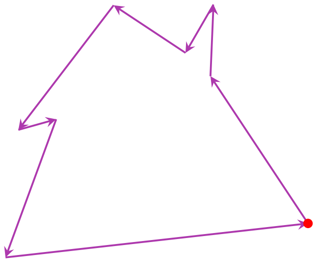


$=$

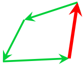
[Guibas et al. (1983)]

Convolution approach

$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1}) \\
P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j
\end{aligned}
$$



$\otimes$

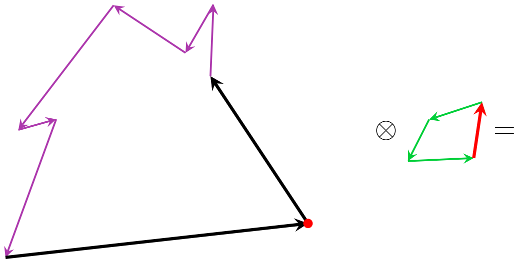


$=$

[Guibas et al. (1983)]

Convolution approach

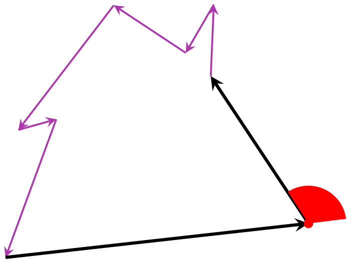
$$\begin{array}{rcl} P & = & (p_0, p_1, \ldots, p_{n-1}) \\ Q & = & (q_0, q_1, \ldots, q_{m-1}) \\ P \otimes Q & = & \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup \end{array}$$



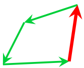
[Guibas et al. (1983)]

Convolution approach

$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1}) \\
P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup
\end{aligned}
$$



$\otimes$

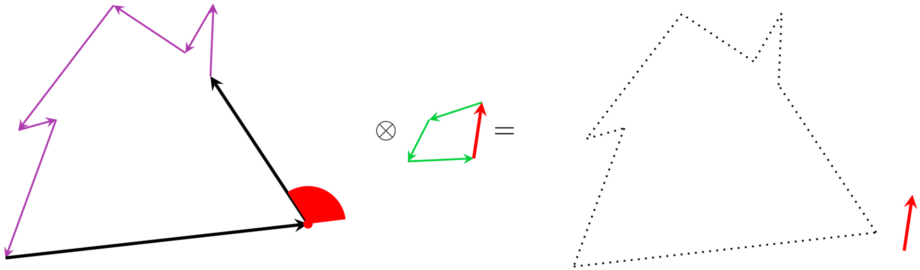


$=$

[Guibas et al. (1983)]

Convolution approach

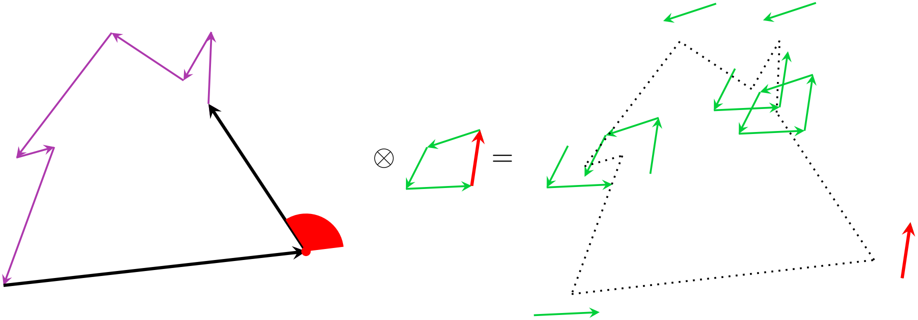
$$\begin{aligned} P &= (p_0, p_1, \ldots, p_{n-1}) \\ Q &= (q_0, q_1, \ldots, q_{m-1}) \\ P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup \end{aligned}$$



[Guibas et al. (1983)]

Convolution approach

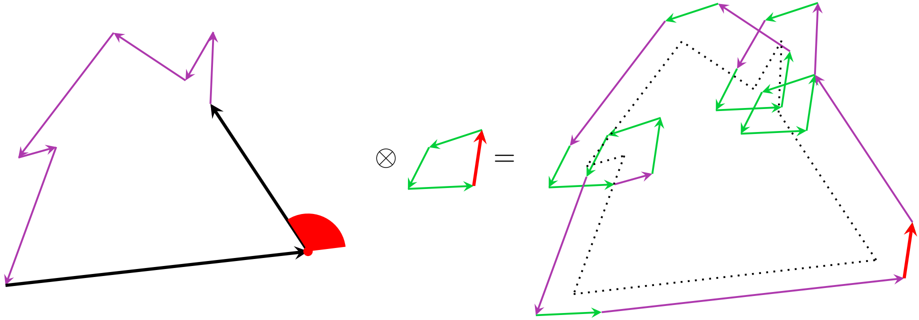
$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1}) \\
P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup
\end{aligned}
$$



[Guibas et al. (1983)]

Convolution approach

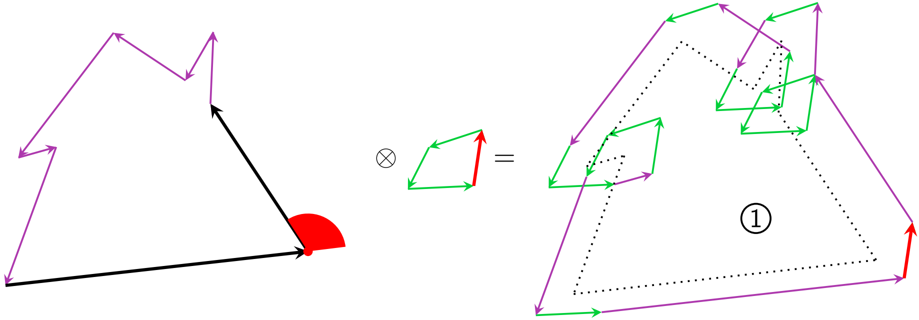
$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1}) \\
P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup \\
& \quad \{\overrightarrow{p_i p_{i+1}} \oplus q_j \mid \overrightarrow{p_i p_{i+1}} \text{ ccw between } \overrightarrow{q_{j-1} q_j} \text{ and } \overrightarrow{q_j q_{j+1}}\}
\end{aligned}
$$



[Guibas et al. (1983)]

Convolution approach

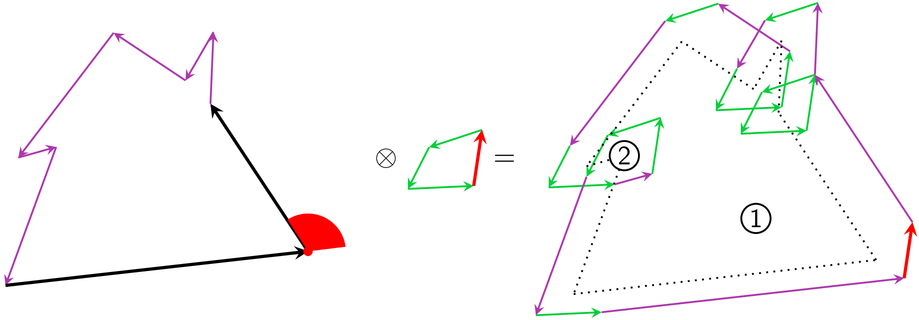
$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1}) \\
P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup \\
&\quad\ \{\overrightarrow{p_i p_{i+1}} \oplus q_j \mid \overrightarrow{p_i p_{i+1}} \text{ ccw between } \overrightarrow{q_{j-1} q_j} \text{ and } \overrightarrow{q_j q_{j+1}}\}
\end{aligned}
$$



[Guibas et al. (1983)]

Convolution approach

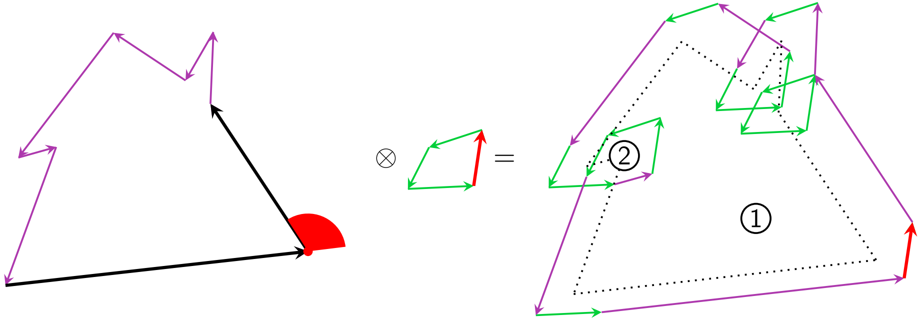
$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1}) \\
P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup \\
& \{\overrightarrow{p_i p_{i+1}} \oplus q_j \mid \overrightarrow{p_i p_{i+1}} \text{ ccw between } \overrightarrow{q_{j-1} q_j} \text{ and } \overrightarrow{q_j q_{j+1}}\}
\end{aligned}
$$



[Guibas et al. (1983)]

Convolution approach

$$
\begin{aligned}
P &= (p_0, p_1, \ldots, p_{n-1}) \\
Q &= (q_0, q_1, \ldots, q_{m-1}) \\
P \otimes Q &= \{\overrightarrow{q_i q_{i+1}} \oplus p_j \mid \overrightarrow{q_i q_{i+1}} \text{ ccw between } \overrightarrow{p_{j-1} p_j} \text{ and } \overrightarrow{p_j p_{j+1}}\} \quad \cup \\
& \{\overrightarrow{p_i p_{i+1}} \oplus q_j \mid \overrightarrow{p_i p_{i+1}} \text{ ccw between } \overrightarrow{q_{j-1} q_j} \text{ and } \overrightarrow{q_j q_{j+1}}\}
\end{aligned}
$$



Bottleneck: arrangement computation, worst case size $O(n^2 m^2)$

[Guibas et al. (1983)]

Overview

1. Can we speed up the convolution approach?
2. Can we fill in holes?
3. How does the algorithm compare to other approaches?

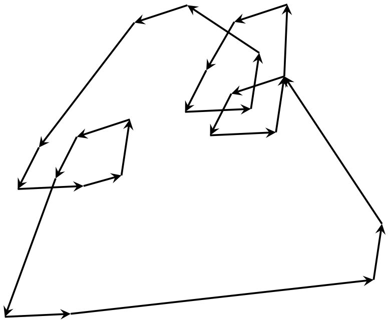
Observation

Reflex vertices don't contribute to the Minkowski Sum's boundary!

[Kaul et al. (1992)]

Observation

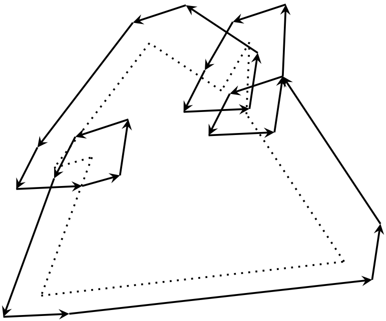
Reflex vertices don't contribute to the Minkowski Sum's boundary!



[Kaul et al. (1992)]

Observation

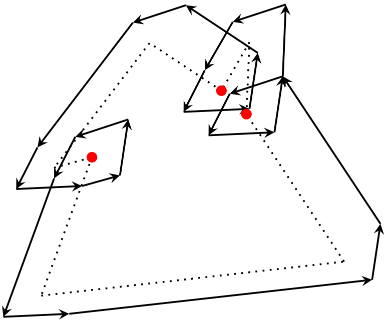
Reflex vertices don't contribute to the Minkowski Sum's boundary!



[Kaul et al. (1992)]

Observation

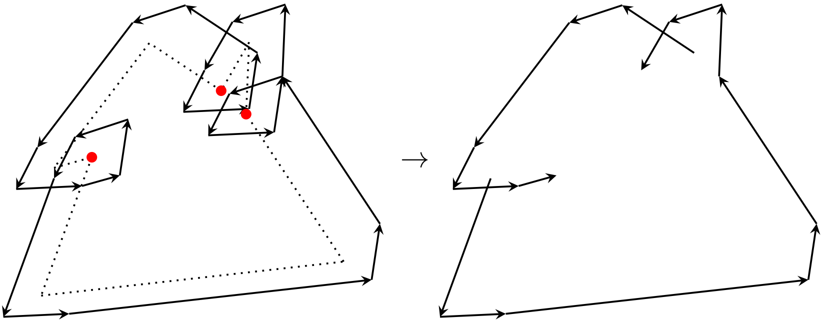
Reflex vertices don't contribute to the Minkowski Sum's boundary!



[Kaul et al. (1992)]

Observation

Reflex vertices don't contribute to the Minkowski Sum's boundary!



[Kaul et al. (1992)]

Reduced convolution approach

Step 1

Compute the reduced convolution.

[Behar and Lien, 2011]

Reduced convolution approach

Step 1

Compute the reduced convolution.

Step 2

Compute the arrangement of the segments.

[Behar and Lien, 2011]

Reduced convolution approach

Step 1

Compute the reduced convolution.

Step 2

Compute the arrangement of the segments.

The winding number property can not be used anymore.

[Behar and Lien, 2011]

Reduced convolution approach

Step 1

Compute the reduced convolution.

Step 2

Compute the arrangement of the segments.

The winding number property can not be used anymore.

Instead, two more steps to remove *false holes*.

[Behar and Lien, 2011]

Quick orientation filter

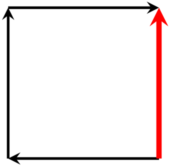
Step 3

Iterate over all faces, discard loops which are not orientable.

Quick orientation filter

Step 3

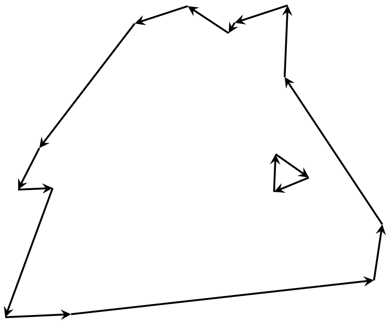
Iterate over all faces, discard loops which are not orientable.



Step 4

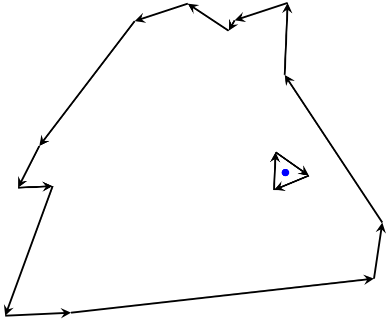
Step 4

For all remaining faces:



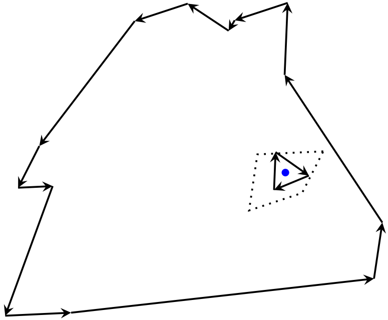
Step 4

For all remaining faces: Find a point inside the face,



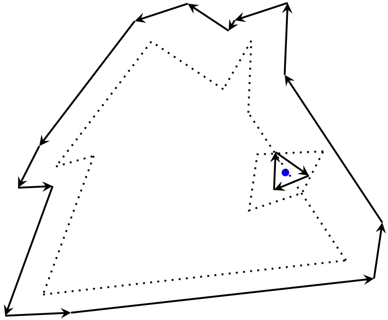
Step 4

For all remaining faces: Find a point inside the face, translate $-Q$ to that point,



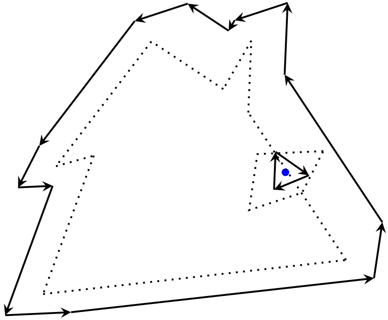
Step 4

For all remaining faces: Find a point inside the face, translate $-Q$ to that point, and intersect it with P.



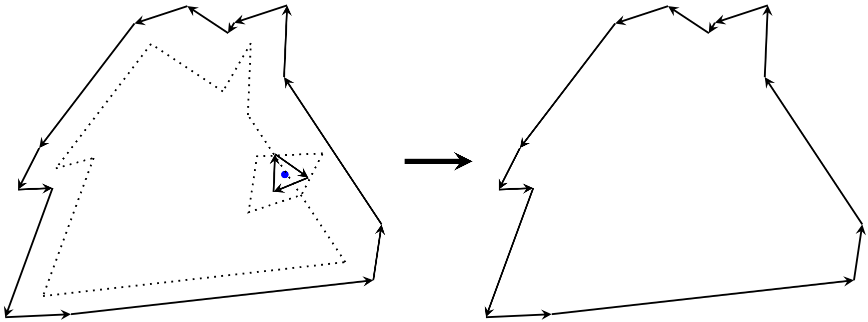
Step 4

For all remaining faces: Find a point inside the face, translate $-Q$ to that point, and intersect it with P. If there is an intersection,



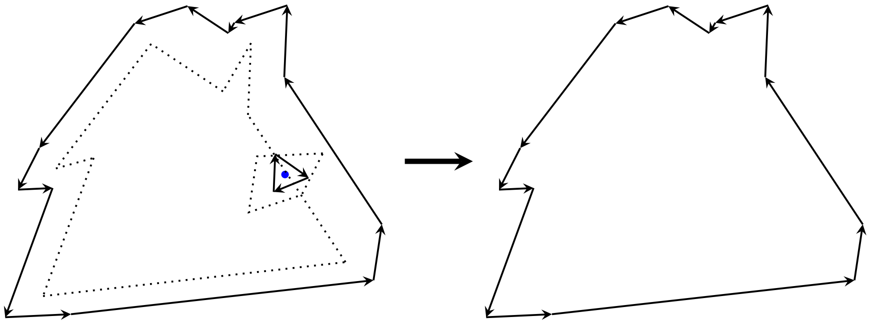
Step 4

For all remaining faces: Find a point inside the face, translate $-Q$ to that point, and intersect it with P. If there is an intersection, discard the face.



Step 4

For all remaining faces: Find a point inside the face, translate $-Q$ to that point, and intersect it with P. If there is an intersection, discard the face.



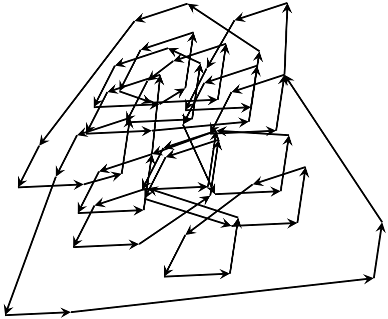
The remaining faces are exactly the holes of $P \oplus Q$!

What about holes?

Even the reduced convolution can become quite complex:

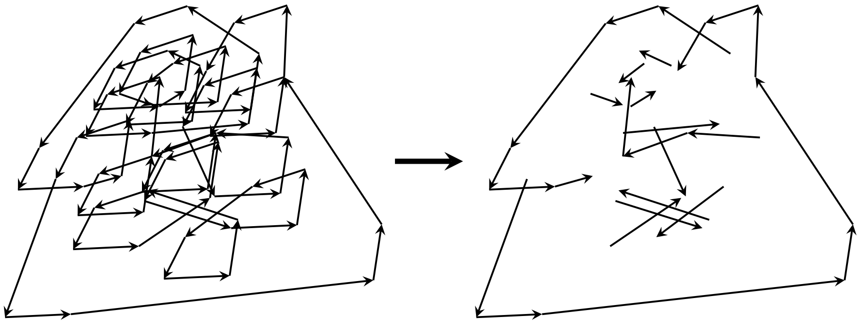
What about holes?

Even the reduced convolution can become quite complex:



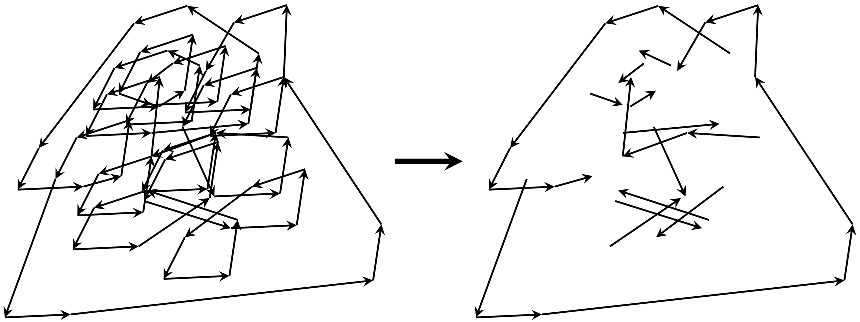
What about holes?

Even the reduced convolution can become quite complex:



What about holes?

Even the reduced convolution can become quite complex:



Idea: Reduce input complexity!

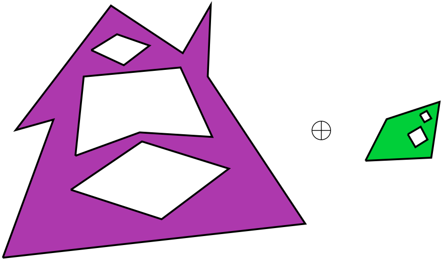
Overview

1. Can we speed up the convolution approach?
2. Can we fill in holes?
3. How does the algorithm compare to other approaches?

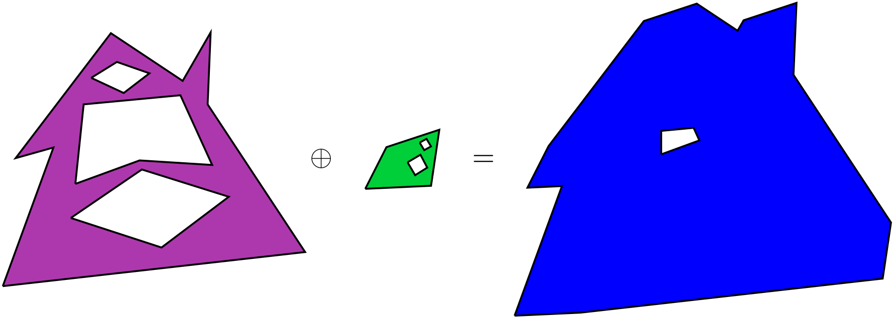
First, we need some polygons with holes. . .

First, we need some polygons with holes. . .

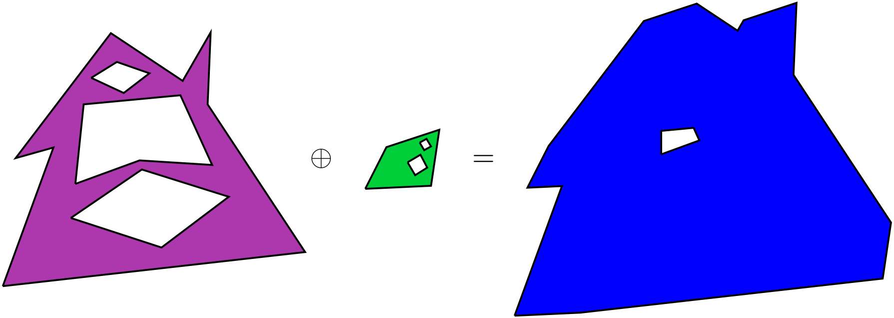
Observation



Observation



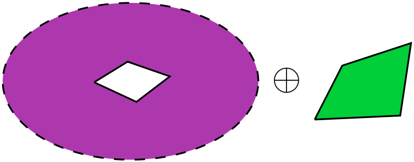
Observation



Which holes are relevant?

Hole filter

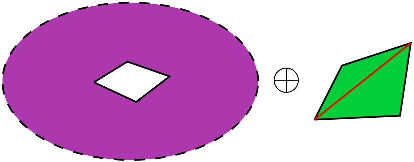
Theorem



Hole filter

Theorem

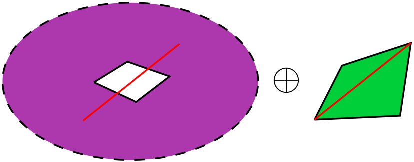
If there is a path γ in Q



Hole filter

Theorem

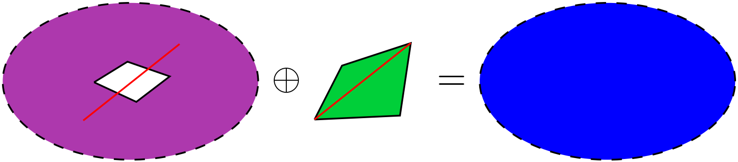
If there is a path γ in Q so that $-\gamma$ does not fit under any translation inside a hole of P,



Hole filter

Theorem

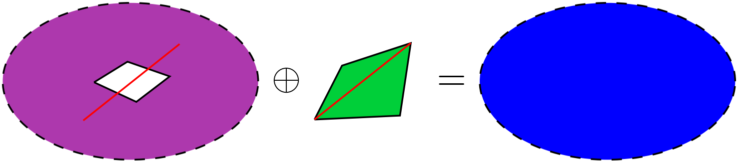
If there is a path γ in Q so that $-\gamma$ does not fit under any translation inside a hole of P, then that hole can be filled up.



Hole filter

Theorem

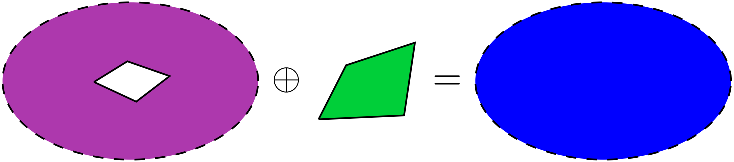
If there is a path γ in Q so that $-\gamma$ does not fit under any translation inside a hole of P, then that hole can be filled up.



Because when the hole's boundary is added to γ, it "smears" completely over the hole.

Corollaries

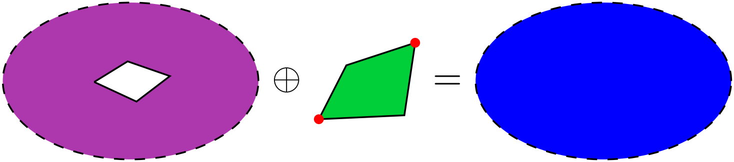
Corollary



Corollaries

Corollary

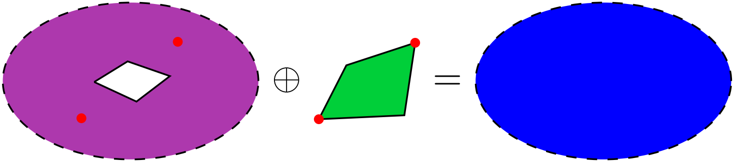
If there are two points in Q



Corollaries

Corollary

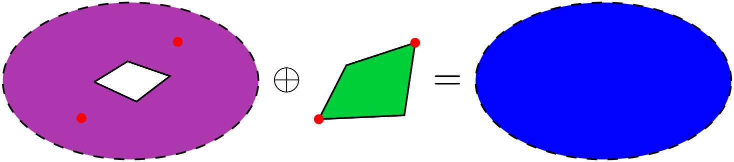
If there are two points in Q so that their inverse does not fit under any translation inside a hole of P,



Corollaries

Corollary

If there are two points in Q so that their inverse does not fit under any translation inside a hole of P, then that hole can be filled up.



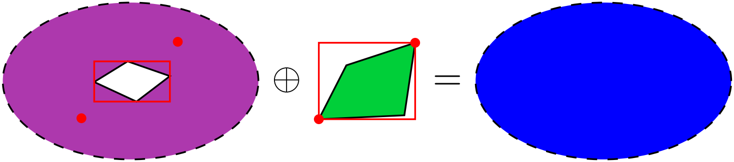
Corollaries

Corollary

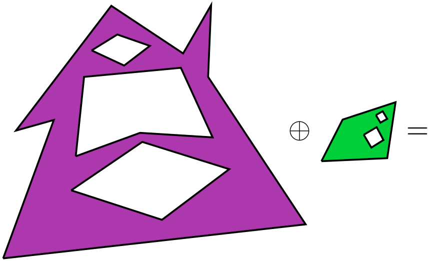
If there are two points in Q so that their inverse does not fit under any translation inside a hole of P, then that hole can be filled up.

Corollary

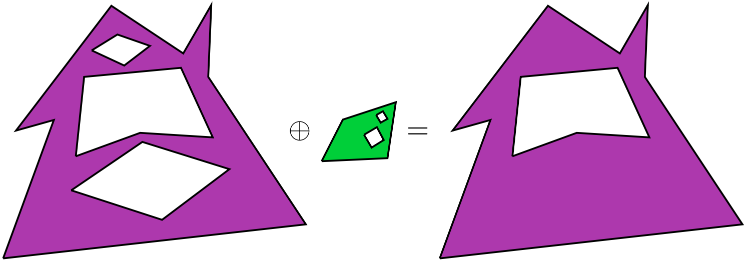
If Q's axis-aligned bounding box does not completely fit inside the hole's axis-aligned bounding box, the hole can be filled up.



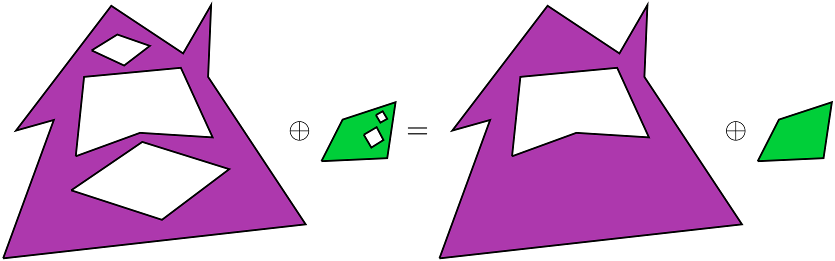
Effect on input



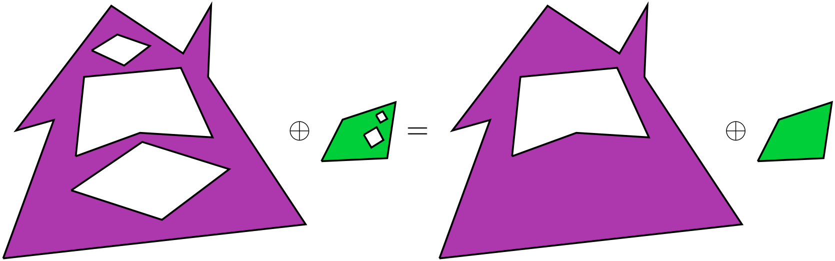
Effect on input



Effect on input

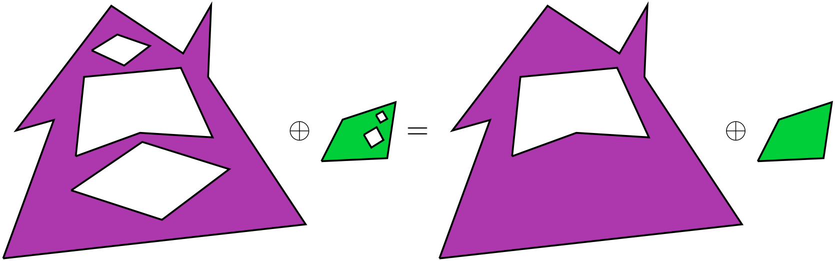


Effect on input



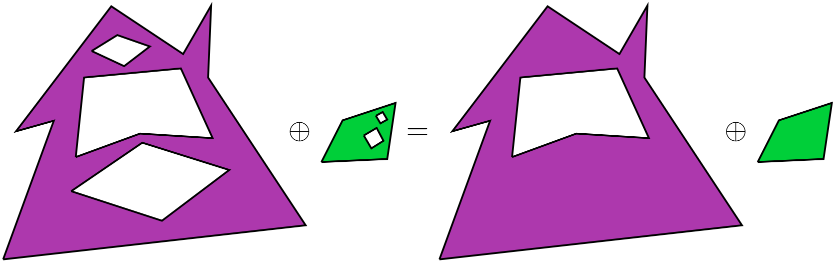
- One is always simple!

Effect on input



- One is always simple! Sometimes both.

Effect on input

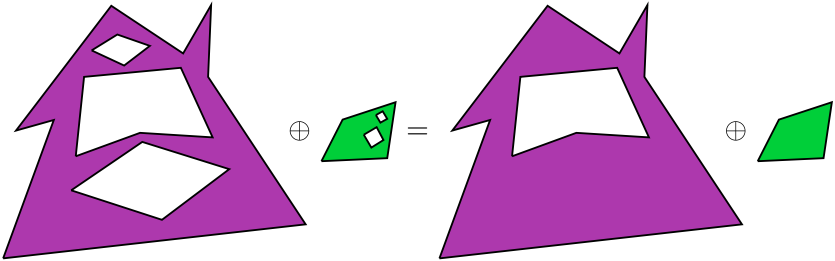


- One is always simple! Sometimes both.

This filter is...

- approach independent

Effect on input



- One is always simple! Sometimes both.

This filter is...

- approach independent
- generalizable to higher dimensions

Overview

1. Can we speed up the convolution approach?
2. Can we fill in holes?
3. How does the algorithm compare to other approaches?

Implementation

	Simple polygons	**General polygons**
Inexact		
Exact		

Implementation

	Simple polygons	General polygons
Inexact		
Exact	Wein (2006)	

Implementation

	Simple polygons	General polygons
Inexact		Behar and Lien (2011)
Exact	Wein (2006)	

Implementation

	Simple polygons	**General polygons**
Inexact		Behar and Lien (2011)
Exact	Wein (2006)	**NEW**

Implementation

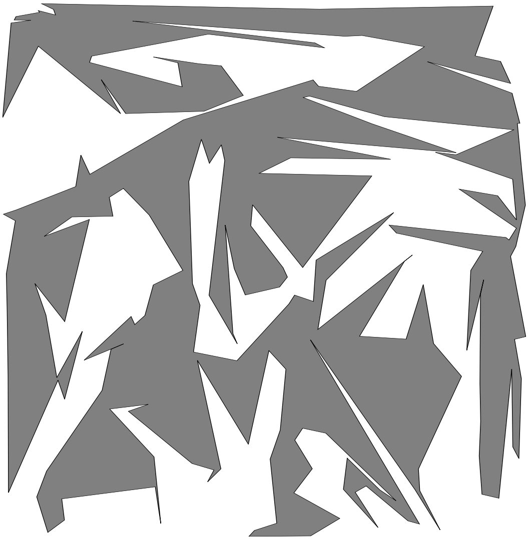
	Simple polygons	**General polygons**
Inexact		Behar and Lien (2011)
Exact	Wein (2006)	**NEW**

Implementation

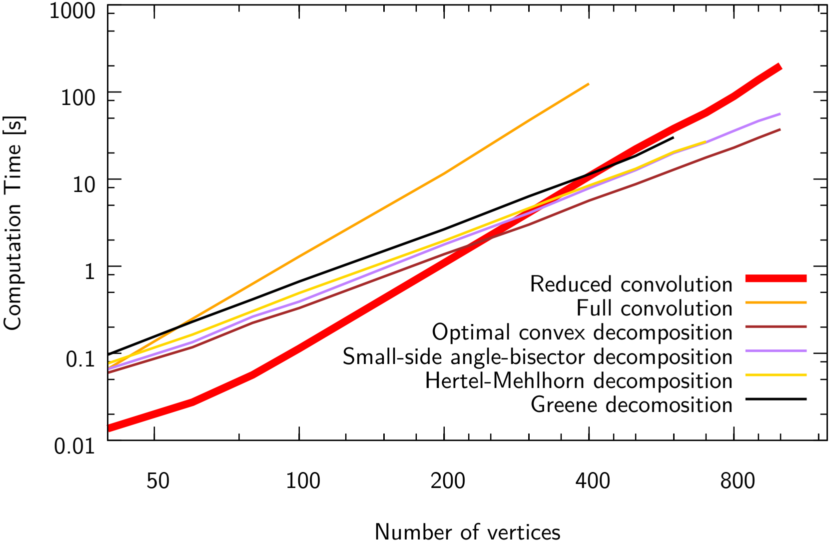
	Simple polygons	General polygons
Inexact		Behar and Lien (2011)
Exact	Wein (2006)	**NEW**

Starting with CGAL 4.7, you can use the method `CGAL::minkowski_sum_2()` on polygons with holes!

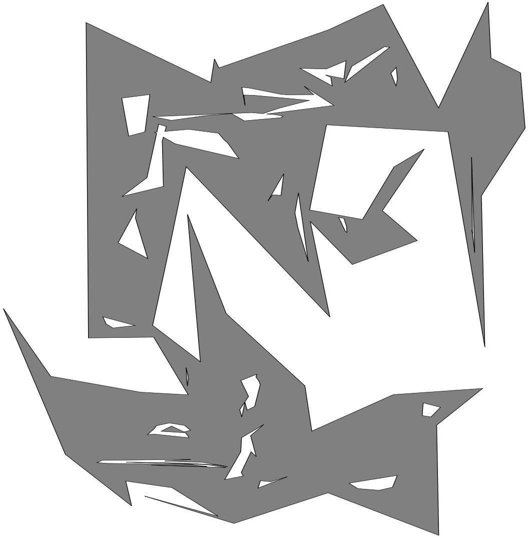
Benchmark: Simple polygons



Benchmark: Simple polygons



Benchmark: Polygons with holes



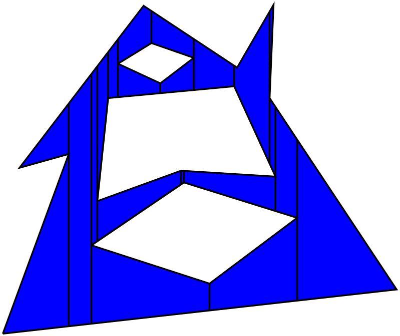
Implemented decomposition approaches

Vertical decomposition

Constrained triangulation

Implemented decomposition approaches

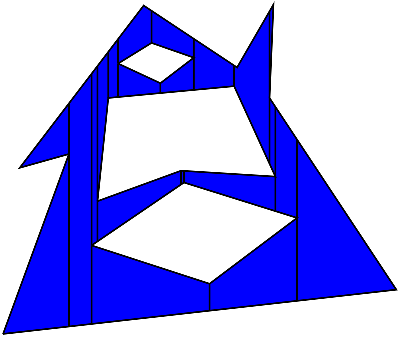
Vertical decomposition



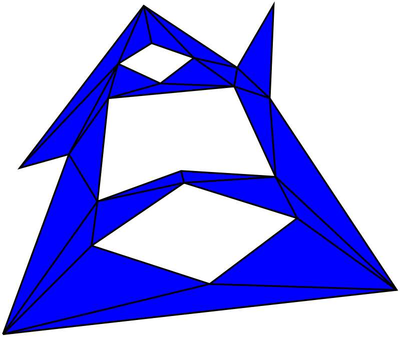
Constrained triangulation

Implemented decomposition approaches

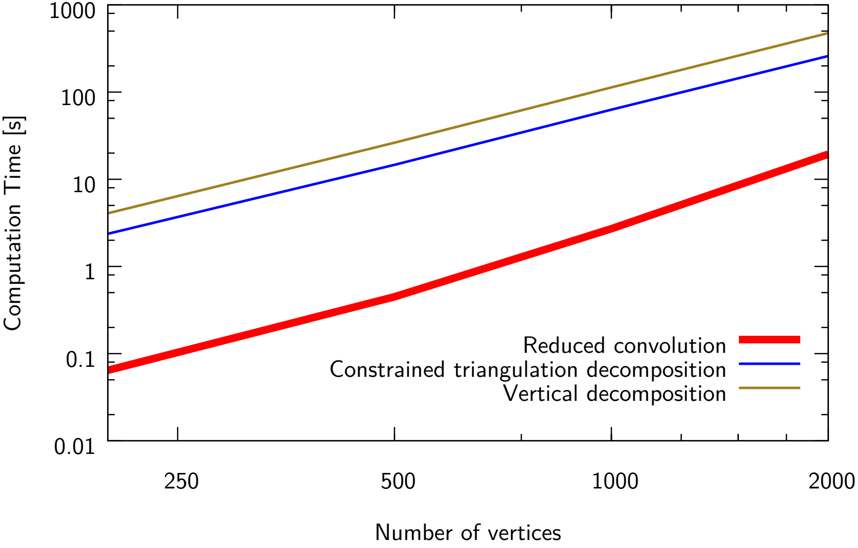
Vertical decomposition



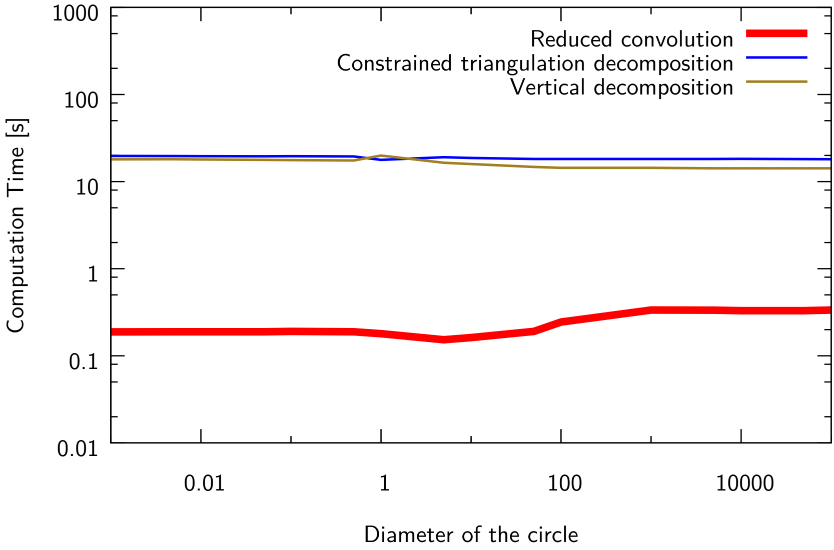
Constrained triangulation



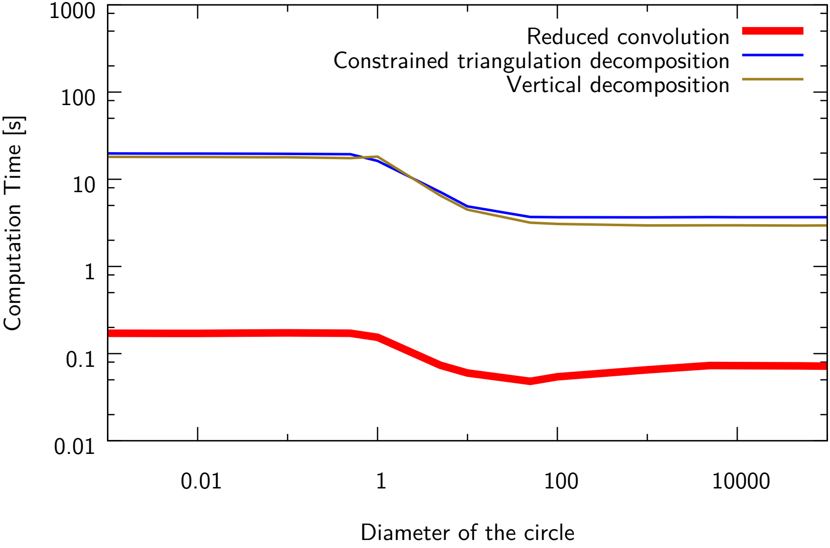
Benchmark: Polygons with holes



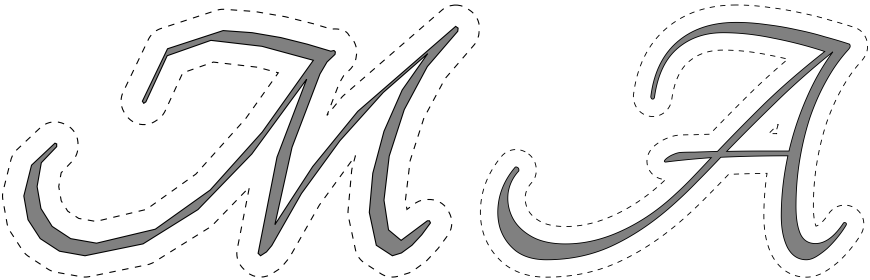
Benchmark: Growing circle (no hole filter)



Benchmark: Growing circle (with hole filter)

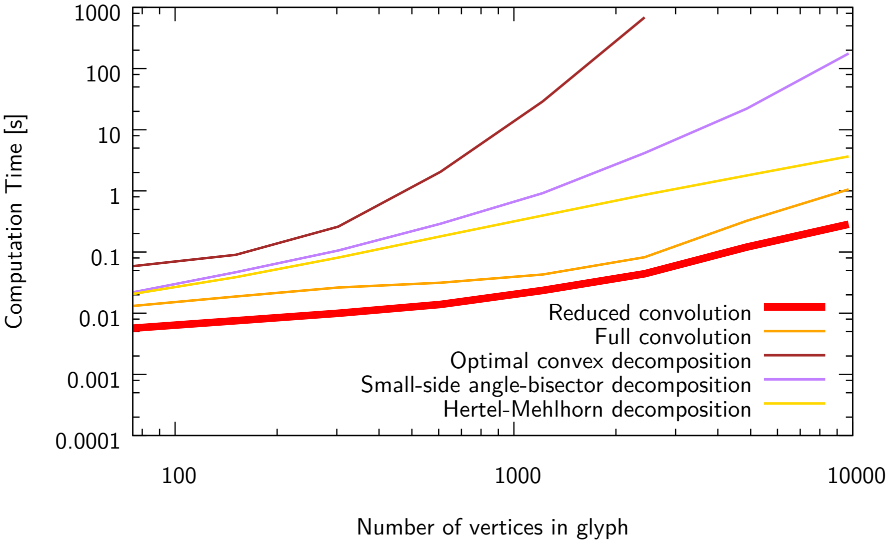


Benchmark: Glyph Offset

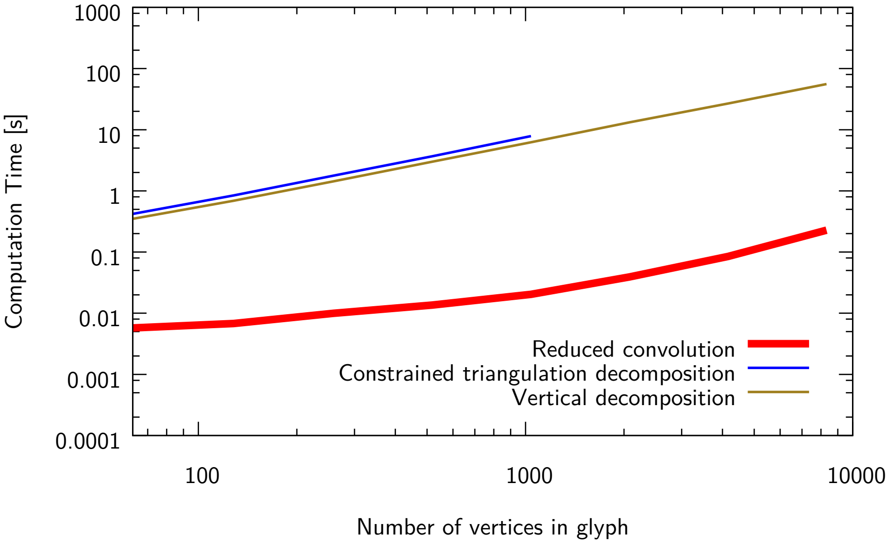


(75 vertices) (8319 vertices)

Benchmark: Glyph Offset (letter M)



Benchmark: Glyph Offset (letter A)



Contributions

1. Minkowski Sum of polygons with holes in CGAL
    - reduced convolution
    - two decomposition methods

Contributions

1. Minkowski Sum of polygons with holes in CGAL
   - reduced convolution
   - two decomposition methods
2. General input-level **hole filter** to reduce complexity

Contributions

1. Minkowski Sum of polygons with holes in CGAL
   - reduced convolution
   - two decomposition methods
2. General input-level **hole filter** to reduce complexity

Thanks!