**Result document**

**2014.12.25**

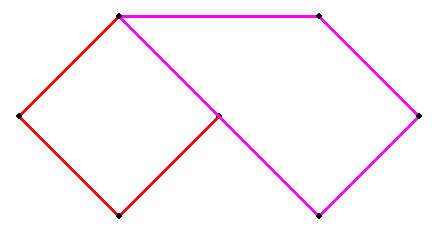
1. **Documentation description**

This program implements three Boolean operations of two-dimensional general polygons such as union, intersection and difference. This document documents the testing of this program, including functional testing and legality checks. Since the implemented algorithm has no advantage over the general Boolean operation algorithm in complexity, no performance test is carried out, and the specific complexity analysis can be found in section 5.2 of the principle document. At the end of the document, some of the more beautiful Boolean results are shown.

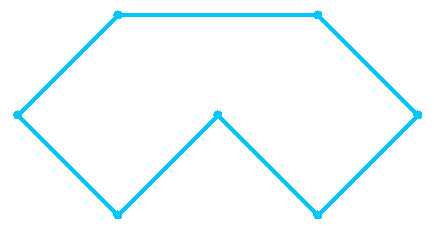
1. **Test**

**2.1 Functional Testing**

A. Two polygon edges coincide

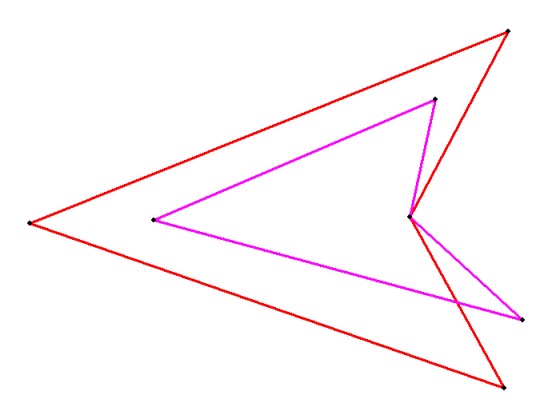


Boolean result of the operation:

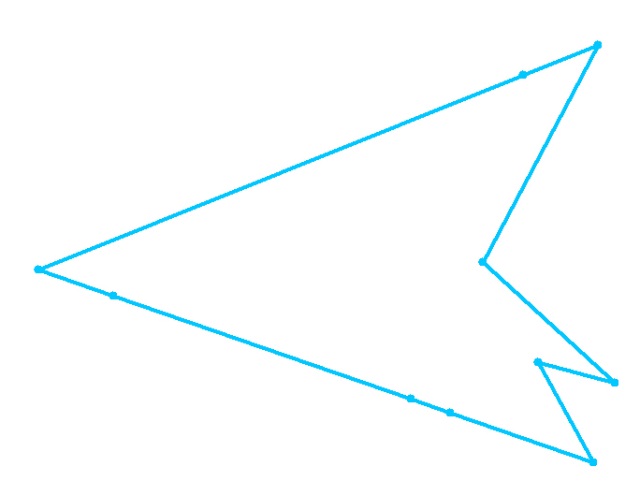
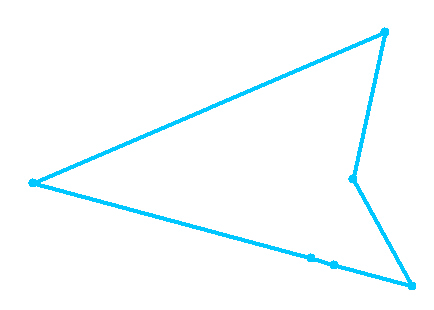


A∪B

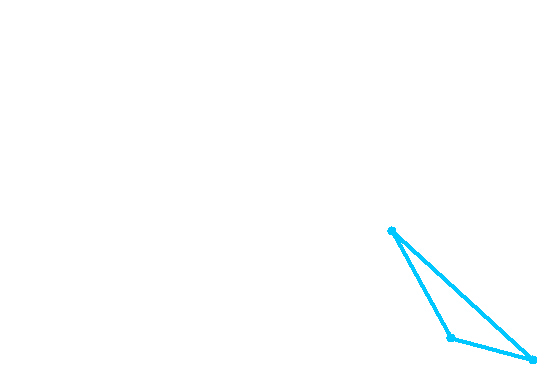
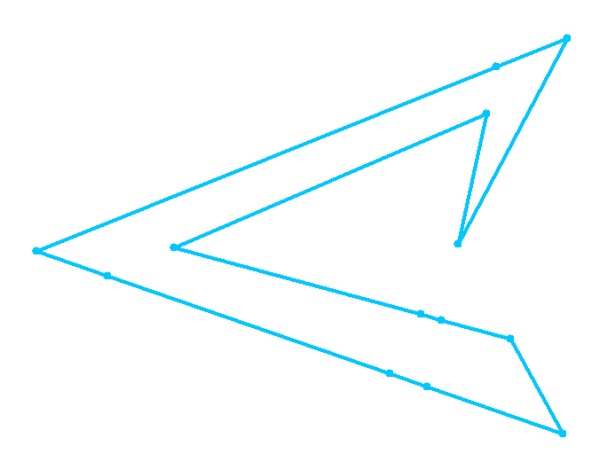
B. The two polygons coincide somewhat



The result of a Boolean operation

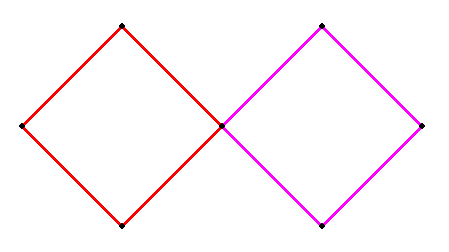
 

A∪B A∩B

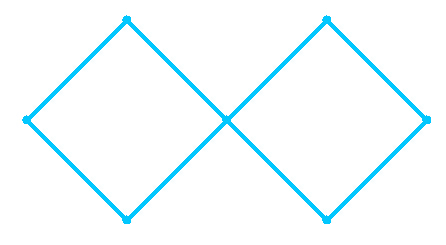


A-B B-A

C. The two multiple deformations coincide somewhat and the edges are collinear

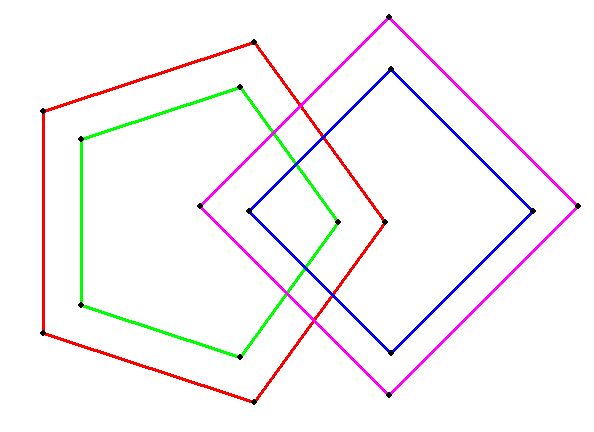


Boolean result of the operation:

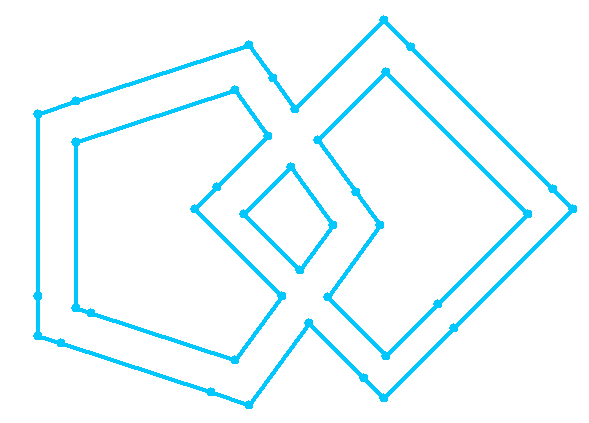
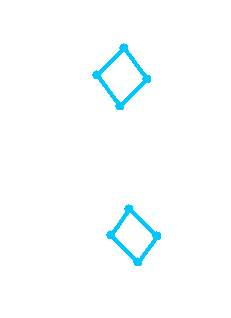


A∪B

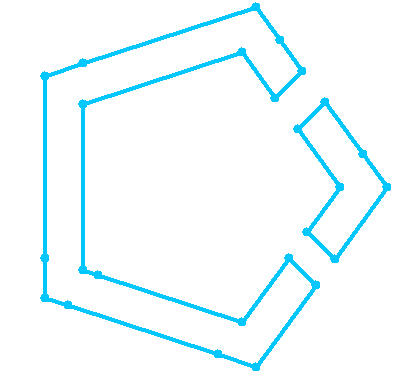
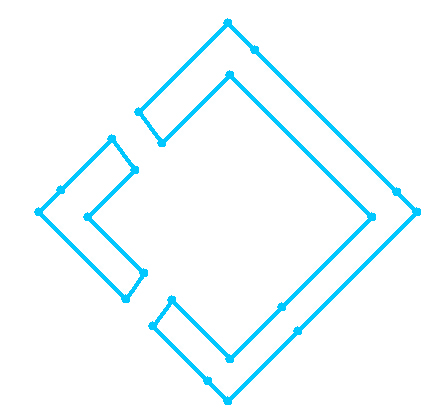
D. Boolean operations between polygons with inner rings



Boolean result of the operation:

A∪B A∩B

A-B B-A

**2.3 Performance Testing**

Scenario 1 in the third major section was tested, where the number of side A was 95 and the number of side B was 68.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | operate | Create BspTree (ms). | Merge(ms) | Generate edges (ms). |
| Before optimization | A∪B | 17 | 22 | 25 |
| A∩B | 17 | 22 | 16 |
| A-B | 16 | 22 | 16 |
| B-A | 17 | 22 | 20 |
| After optimization | A∪B | 6 | 2 | 25 |
| A∩B | 6 | 3 | 17 |
| A-B | 5 | 3 | 16 |
| B-A | 5 | 4 | 20 |

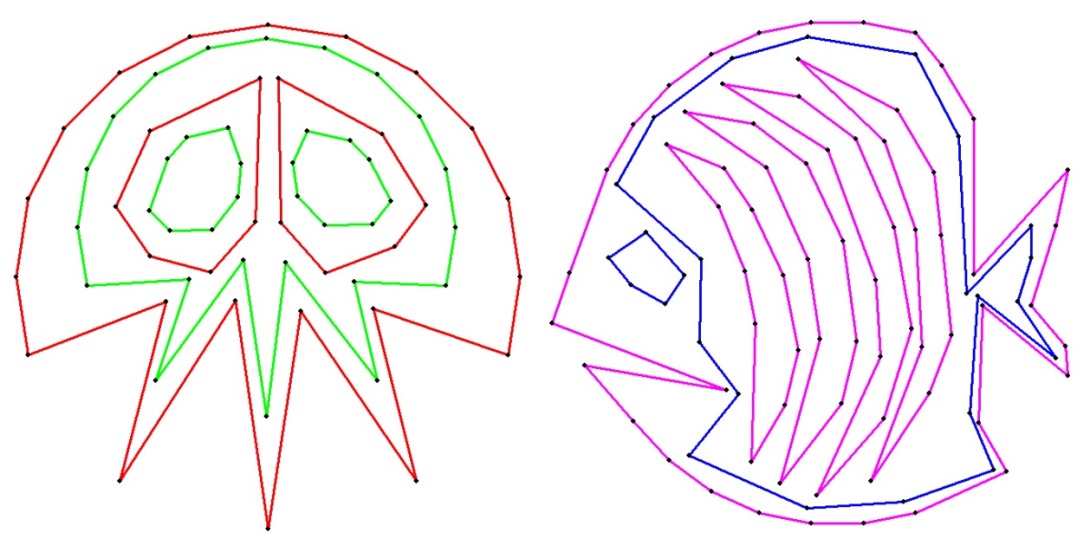
Scenario 2 in the third major section was tested, where the number of side A was 45 and the number of side B was 86.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | operate | Create BspTree (ms). | Merge(ms) | Generate edges (ms). |
| Before optimization | A∪B | 11 | 12 | 11 |
| A∩B | 10 | 13 | 7 |
| A-B | 10 | 13 | 7 |
| B-A | 10 | 29 | 20 |
| After optimization | A∪B | 5 | 1 | 10 |
| A∩B | 4 | 2 | 7 |
| A-B | 5 | 2 | 6 |
| B-A | 4 | 4 | 22 |

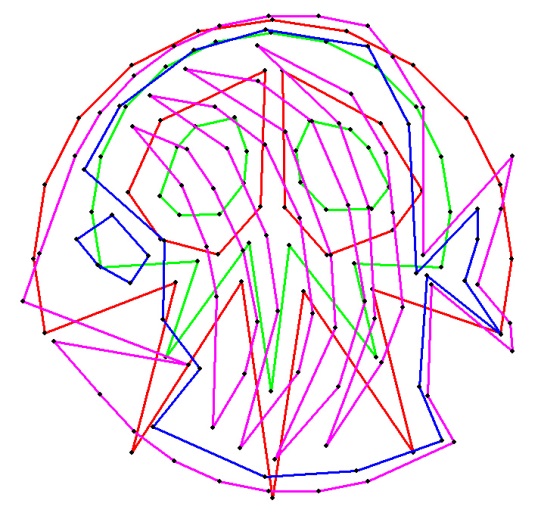
It can be seen that after optimization, both the BspTree settling time and the merge time have dropped significantly.

1. **Case show**

Case 1: Polygon B on the left and Polygon A on the right



Overlapping together:



Boolean result of the operation:

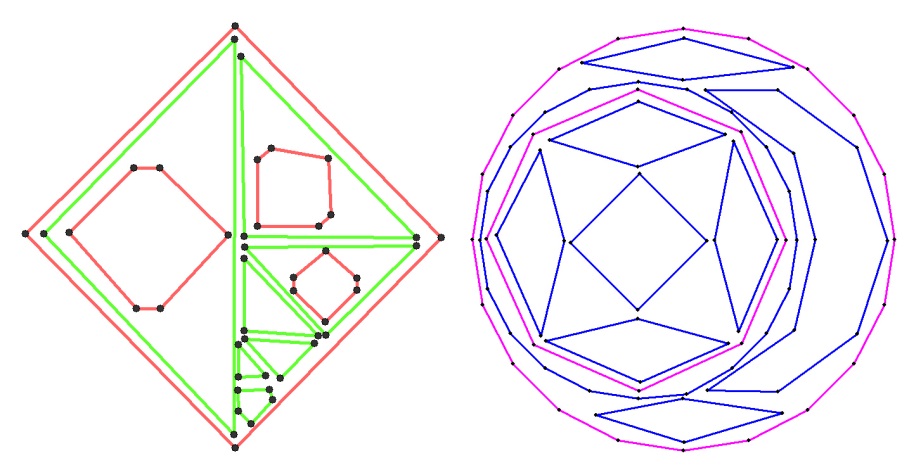
 

A∪B A∩B

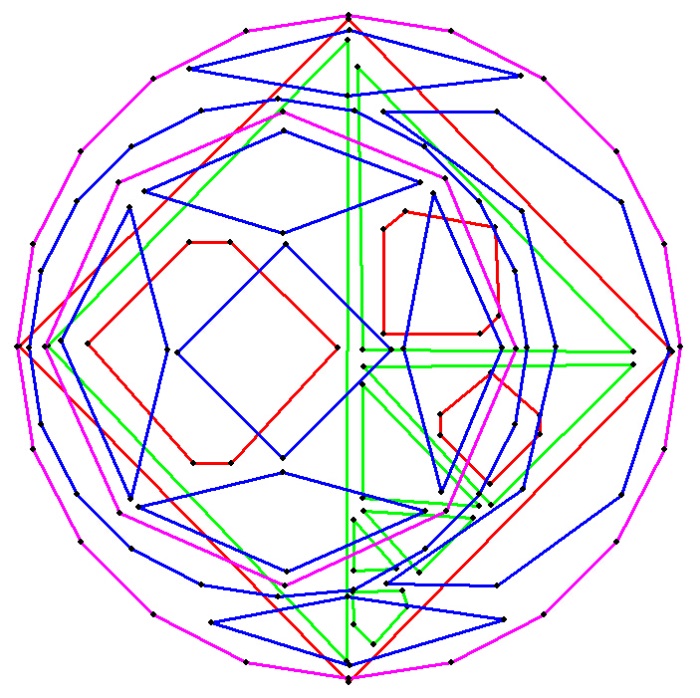
 

A-B B-A

Case 2: Polygon A on the left and polygon B on the right



Overlapping together:



Boolean result of the operation:

A∪B A∩B

A-B B-A