Алгоритм пересечения двух выпуклых многоугольников

Предполагаем, что используется аффинная плоскость, то есть векторы прикладываются не обязательно к началу координат. Для ненулевого вектора a обозначим (a) прямую, имеющую направляющий вектор, параллельный a, и проходящую через начало вектора a (или, что то же самое, проходящую через точки начала и конца вектора a). Для вектора a обозначим $[a]$ отрезок, соединяющий начальную и конечную точки a. Для векторов a и b обозначим $\langle a, b\rangle$ их скалярное произведение: $\langle a, b\rangle = a_x b_x + a_y b_y$. Для векторов a и b обозначим $a \wedge b$ их антисимметричное (косое) произведение: $a \wedge b = a_x b_y - a_y b_x$.

Пусть даны два выпуклых многоугольника P и Q с наборами вершин $V(P) = [p_0, p_1, \ldots, p_{n-1}]$ и $V(Q) = [q_0, q_1, \ldots, q_{k-1}]$, заданные цикличными списками своих вершин в порядке обхода против часовой стрелки. Полагаем, что в многоугольниках все внутренние углы *строго* меньше $180°$.

Пусть $p \in V(P)$, $q \in V(Q)$ произвольные вершины P и Q, соответственно. Обозначим p_- точку предшествующую p, а p_+ следующую за p в циклическом порядке. Вектор из p_- в p обозначим $\hat{p}$. Для q аналогично.

$H^+(\hat{p})$ — открытая левая полуплоскость, задаваемая вектором $\hat{p}$:

$$H^+(\hat{p}) = \{x \in \mathbb{R}^2 \mid \hat{p} \wedge (x - p_-) > 0\}.$$

Если выполняются общие предположения, то правила перемещения векторов описываются следующей таблицей:

$\hat{p} \wedge \hat{q}$	Условие полуплоскости	Двигаем
$\geqslant 0$	$q \in H^+(\hat{p})$	p
$\geqslant 0$	$q \notin H^+(\hat{p})$	q
< 0	$p \in H^+(\hat{q})$	q
< 0	$p \notin H^+(\hat{q})$	p

Алгоритм 1 Алгоритм пересечения двух выпуклых многоугольников

Вход: P и Q
Выход: $R = P \cap Q$
Выбрать произвольно $p \in V(P)$ и $q \in V(Q)$
$V(R) \leftarrow \varnothing$
`inside` $\leftarrow$ `unknown`
повторять
    (`crossType`, `point`) $\leftarrow$ Пересечь($[\hat{p}]$, $[\hat{q}]$)
    если `crossType` $\equiv$ *SinglePoint* **тогда**
        если `point` совпадает с первой точкой в $V(R)$ **тогда**
            закончить алгоритм
        иначе если $p \in H^+(\hat{q})$ **тогда** `inside` $\leftarrow$ `inP`
        иначе если $q \in H^+(\hat{p})$ **тогда** `inside` $\leftarrow$ `inQ`
            ▷ иначе `inside` не меняется.
        конец если
        Добавить(`point`)
    конец если

    `cross` $\leftarrow$ $\mathrm{sign}(\hat{p} \wedge \hat{q}\,)$
    если `crossType` $\equiv$ *Overlap* И $\langle \hat{p}, \hat{q}\,\rangle < 0$ **тогда**
        $V(R) \leftarrow \varnothing$
        закончить алгоритм
    иначе если `crossType` $\equiv$ *Overlap*
        Двигать(p, ЛОЖЬ).
    иначе если `cross` $\geqslant 0$
        если $q \in H^+(\hat{p})$ **тогда**
            Двигать(p, `inside` $\equiv$ `inP`)
        иначе
            Двигать(q, `inside` $\equiv$ `inQ`)
        конец если
    иначе ▷ если `cross` < 0
        если $p \in H^+(\hat{q})$ **тогда**
            Двигать(q, `inside` $\equiv$ `inQ`)
        иначе
            Двигать(p, `inside` $\equiv$ `inP`)
        конец если
    конец если
пока не выполняется: Оба списка пройдены ИЛИ один из них пройден два раза
Если $V(R) \equiv \varnothing$, то обработать случаи, $P \subset Q$, $Q \subset P$, $R \equiv \varnothing$. Делается за логарифм двоичным поиском по углу.

Процедура Добавить(Вершина v)

  если $V(R) \equiv \varnothing$ ИЛИ $v \neq$ последней точке в $V(R)$ **тогда**

    $V(R) \leftarrow v$

  конец если

Конец процедуры

Процедура Двигать(Вершина v, флаг_Выводить)

  если флаг_Выводить **тогда**

    Добавить(v)

  конец если

  перейти к следующей вершине многоугольника, соответствующего вершине v.

Конец процедуры