Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

# АЛГОРИТМЫ ОТСЕЧЕНИЯ ОТРЕЗКОВ И МНОГОУГОЛЬНИКОВ В ОДНОРОДНЫХ КООРДИНАТАХ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Андрияновой Марии Игоревны

Научный руководитель
Зав. кафедрой, к. ф.-м. н. _____ С. В. Миронов
10.06.17

Заведующий кафедрой
к.ф.-м.н. _____ С. В. Миронов
10.06.17

Саратов 2017

**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

На данный момент не существует единого универсального подхода к решению проблемы отсечения отрезков и многоугольников. Широко используемые для отсечения алгоритмы формулируются в терминах вещественнозначных вычислений, а это при реализации требует значительных временных и емкостных затрат. Но алгоритмов отсечения, сформулированных в целочисленных координатах, на сегодняшний день не представлено, поэтому задача получения таких алгоритмов остается актуальной.

Из-за своей специфики аппарат однородных координат позволяет находить точки пересечения отрезков в целочисленных координатах, при условии, что исходные координаты также являются целочисленными [1]. Так как большинство графических систем предполагает целочисленность координат точек, то представляется разумным применение аппарата однородных координат в задачах отсечения отрезков прямых и многоугольников.

Цель выпускной квалификационной работы — получить методы отсечения прямых и многоугольников, сформулированные в теминах однородных координат и провести сравнительный анализ реализованных алгоритмов с известными алгоритмами отсечения, оперирующими в евклидовом пространстве.

Данная работа состоит из двух глав. В первой главе «Алгоритмы компьютерной графики для отсечения отрезков и многоугольников» излагается перечень областей применения компьютерной графики, раскрывается идея задачи отсечения, а также приводится описание наиболее популярных алгоритмов отсечения отрезков и многоугольников. Во второй главе «Алгоритмы отсечения линий и многоугольников в однородных координатах» обосновывается применение аппарата однородных координат в алгоритмах отсечения, описываются разработанные методы отсечения отрезков и многоугольников в однородных координатах. Эта глава также включает информацию о входных файлах, которые были сгенерированы для проведения тестирования алгоритмов. В конце главы представляются результаты тестирования и сравнительный анализ известных алгоритмов компьютерной графики (алгоритм Скала [2] и алгоритм Сазерленда—Ходгмана [2–4]) и разработанных алгоритмов отсечения отрезков и многоугольников.

## 1 Получение метода отсечения отрезков

За основу разрабатываемого метода отсечения отрезков возьмем алгоритм Скала.

Будем предполагать, что все координаты исходных точек в этом методе заданы в однородных координатах.

Для каждого ребра $F_iF_{i+1}$ области видимости находим коэффициенты уравнения несущей прямой. Обозначим такую прямую через $l_i$. Так как имеются однородные координаты точки $F_i$ и однородные координаты точки $F_{i+1}$, то однородные координаты прямой $l_i$ можно найти через векторное произведение $F_i \times F_{i+1}$.

Известно, что любая прямая делит пространство на 2 части: положительную и отрицательную. Многоугольник, относительно которого будет происходить отсечение, является выпуклым, поэтому все остальные его вершины должны лежать по одну сторону от прямой $l_i$. Для всех прямых, ограничивающих многоугольник области видимости, будем выбирать такую полярность, чтобы все его вершины, не лежащие на этой прямой, находились в положительном полупространстве. Так, для этого берем вершину $F_{i+2}$ и проверяем скалярное произведение $l_iF_{i+2}$. Если оно меньше нуля, то вместо прямой $l_i$ берем прямую $-l_i$, то есть домножаем все координаты прямой $l_i$ на $-1$, таким образом меняя полярность.

На вход алгоритма подается список отрезков для последующего отсечения. Рассматриваем очередной отрезок $AB$. Имея однородные координаты точки $A$ и однородные координаты точки $B$ через векторное произведение $A \times B$ можно вычислить коэффициенты уравнения несущей прямой, которую обозначим как $l_{ab}$.

Теперь опишем саму идею метода.

Для каждого ребра найдено $l_i$, а также вычислено $l_{ab}$. Найдем скалярные произведения $l_{ab}F_i$ и $l_{ab}F_{i+1}$. После этого перемножим их, и если результат их произведения будет меньше нуля, то скалярные произведения имеют разные знаки, а это значит, что прямая $l_{ab}$ пересекает $F_iF_{i+1}$.

В ходе алгоритма ведется подсчет количества тех ребер многоугольника, для которых найдется точка пересечения с прямой $l_{ab}$. После того, как мы узнали, что прямая $l_{ab}$ пересекает $F_iF_{i+1}$, счетчик нужно увеличить.

Теперь нужно узнать, пересекает ли отрезок $AB$ ребро $F_iF_{i+1}$.

— Если скалярное произведение $l_iA$, умноженное на скалярное произведение $l_iB$, будет меньше либо равно нулю, то точки $A$ и $B$ лежат по разные стороны от прямой $l_i$, следовательно, $AB$ перескает $F_iF_{i+1}$. Точку пересечения $C$ будем находить через векторное произведение $l_i \times l_{ab}$. Пусть $[\chi, \gamma, \alpha]$ — однородные координаты точки $C$. Если $\alpha$ меньше нуля, нужно провести нормализацию. Для этого домножаем все координаты точки $C$ на $-1$.

  Теперь нужно узнать, какой отрезок переходит на следующий этап: $AC$ или $CB$. Если скалярное произведение $l_iA$ меньше нуля, то $A$ лежит на стороне внешности, вместо нее берем точку $C$. Тогда на следующий этап переходит отрезок $CB$, а иначе отрезок $AC$.

— Если скалярное произведение $l_iA$, умноженное на скалярное произведение $l_iB$, окажется больше нуля, то есть эти скалярные произведения имели один и тот же знак, то нам нужно узнать, чему равняется $l_iA$. Если это произведение меньше нуля, то отрезок полностью отбрасывается, если же больше либо равно нулю, то отрезок полностью принимается и переходит на следующий этап.

  Когда нашлись два ребра области видимости, которые пересекаются с $l_{ab}$, мы делаем выход из алгоритма.

  После того, как отрезок принимается, он добавляется в результирующий список отрезков, который и будет выдан в качестве результата работы алгоритма.

## 2 Получение метода отсечения многоугольников

За основу разрабатываемого метода возьмем алгоритм Сазерленда—Ходгмана.

Находим коэффициенты уравнения несущей прямой для каждого ребра области видимости $F_iF_{i+1}$ через векторное произведение $F_i \times F_{i+1}$. Обозначим такую прямую через $l_i$.

Многоугольник отсечения является выпуклым, поэтому все остальные его вершины должны лежать по положительную сторону от прямой $l_i$. Для этого делаем проверку: берем вершину $F_{i+2}$ и находим скалярное произведение $l_iF_{i+2}$, и если оно меньше нуля, то вместо прямой $l_i$ берем прямую $-l_i$, то есть домножаем все координаты прямой $l_i$ на $-1$, таким образом меняя полярность.

На вход алгоритма подается список многоугольников. Берем очередной многоугольник из этого списка. Берем очередную пару вершин $V_j$ и $V_{j+1}$, где $j = 1, ..., M-1$, $M$ — количество вершин текущего отсекаемого многоугольника. Обозначим несущую прямую, на которой лежит отрезок $V_jV_{j+1}$ через $l_{v_jv_{j+1}}$. Найдем коэффициенты уравнения этой прямой через векторное произведение $V_j \times V_{j+1}$.

Теперь рассмотрим сам алгоритм. Для определения расположения точек $V_j$ и $V_{j+1}$ относительно области видимости находим скалярные произведения $l_iV_j$ и $l_iV_{j+1}$. Обозначим такие произведения через $Q_1$ и $Q_2$ соответственно. Перемножим $Q_1$ и $Q_2$. Если полученное произведение меньше нуля, значит эти скалярные произведения имели разные знаки, а это значит, что есть точка пересечения. Точку пересечения $C$ будем находить через векторное произведение $l_i \times l_{v_jv_{j+1}}$. Пусть $[\chi, \gamma, \alpha]$ — однородные координаты точки $C$. Если $\alpha$ меньше нуля, нужно провести нормализацию. Для этого домножаем все координаты точки $C$ на $-1$. После этого точка $C$ добавляется в новый многоугольник — результат отсечения.

Если $Q_2 > 0$, то поворот от $F_iF_{i+1}$ к $F_iV_{j+1}$ осуществляется по часовой стрелке, следовательно, точка $V_{j+1}$ находится внутри области видимости. Если $Q_2 = 0$ — точка $V_{j+1}$ находится на несущей прямой $i$-ого ребра. В этих случаях точка $V_{j+1}$ будет принята, иначе — будет отброшена.

Полученный после отсечения новый многоугольник добавляется в результирующий список многоугольников, который будет выдан в качестве результата работы алгоритма.

## 3 Входные файлы для тестирования алгоритмов

Для проведения тестирования и сравнения стандартного алгоритма Скала отсечения отрезков и модифицированного требуется подготовить файлы со случайными отрезками в следующих вариантах:

1. файлы с разным количеством объектов (100, 200, ..., 1000, 5000, ..., 30000);
2. файлы с различным разбросом, но с фиксированным количеством объектов. Возможны следующие случаи расположения:
   - большое количество объектов внутри области видимости, малое количество — вне области и малое — на границе;
   - большое количество объектов вне области видимости, малое — внутри и на границе области;
   - большое количество объектов на границе области видимости, малое количество — вне и внутри области.

Для тестирования и сравнения стандартного алгоритма Сазерленда—Ходгмана отсечения многоугольников и модифицированного требуется сгенерировать файлы с произвольными многоугольниками. Генерация файлов должна быть произведена следующим образом:

1. файлы с разным количеством многоугольников (100, 200, ..., 1000, 5000, ..., 30000), но все многоугольники с одинаковым числом вершин.
2. файлы с многоугольниками различного вида (треугольниками, четырехугольниками, ..., восьмиугольниками), но с фиксированным количеством.
3. файлы с различным разбросом, но с фиксированным числом объектов, причем все объекты с одинаковым количеством вершин. Были учтены следующие варианты расположения многоугольников относительно области видимости:
   - большое количество объектов внутри области видимости, малое количество — вне области и малое — на границе;
   - большое количество объектов вне области видимости, малое — внутри и на границе области;
   - большое количество объектов на границе области видимости, малое количество — вне и внутри области.

## 4 Реализация программы для генерации входных файлов и проведения тестирования

В рамках настоящей работы была реализована программа в приложении Windows.Forms [5] на языке C++ в среде Microsoft VisualStudio, с помощью которой были получены все необходимые входные данные, проведено тестирование алгоритмов на скорость работы и произведено сравнение по результатам данного тестирования.

Структура файла с полученными случайными координатами отрезков выглядит следующим образом:

```
-494        77        116        -342
-319        249       -373       -139
-463       -169       241         191
-504        443       100        -341
 ...        ...       ...         ...
```

В каждой строке располагаются четыре координаты, первые две — координаты начала отрезка, последние две — координаты конца.

Формат файла с полученными координатами многоугольников выглядит следующим образом:

```
# отсекаемые многоугольники
polygon 3 627 -433 474 -87 -568 -29
polygon 3 68 291 161 322 -347 417
polygon 3 23 165 -417 438 -609 -541
polygon 3 540 -147 -180 -309 -422 344
  ...       ...       ...       ...
```

Сначала идет слово `polygon`, после него указано число, обозначающее количество вершин многоугольника, каждая пара следующих чисел — это координаты $i$-ой вершины многоугольника в порядке обхода вершин по часовой стрелке.

Для проведения теста на скорость была добавлена кнопка «Test». При ее нажатии происходит вызов функции тестирования (выполняются различные манипуляции с изображением: поворот, масштабирование, смещение и т.д.) и вычисляется время работы в секундах.

## 5 Результаты тестирования

В таблице 1 приведены результаты измерения скорости выполнения модифицированного в однородных координатах алгорима отсечения отрезков и алгоритма Скала в зависимости от количества отрезков.

Таблица 1 – Сравнение скорости работы алгоритмов в зависимости от объема входных данных

| Количество объектов | Алгоритм отсечения отрезков в однородных координатах (с) | Алгоритм Скала (с) |
|---|---|---|
| 100 | 0.1090824082 | 0.0784217588 |
| 200 | 0.2191478079 | 0.1542465189 |
| 300 | 0.3145579294 | 0.2306988894 |
| 500 | 0.5011984872 | 0.3790328314 |
| 700 | 0.7295461979 | 0.5215555897 |
| 1000 | 1.0126833083 | 0.7485274038 |
| 5000 | 5.2865688236 | 3.7109803303 |
| 10000 | 10.4041376112 | 7.5135801835 |
| 20000 | 20.3904143123 | 14.3699259698 |
| 30000 | 30.4026203604 | 22.4954319023 |

В таблице 2 показаны результаты теста на скорость для модифицированного в однородных координатах алгоритма отсечения отрезков и алгоритма Скала в зависимости от положения объектов по отношению к области видимости. Количество объектов для каждого случая расположения фиксированное.

Таблица 2 – Сравнение скорости работы алгоритмов в зависимости от разброса объектов относительно области видимости

| Расположение объектов | Алгоритм отсечения отрезков в однородных координатах (с) | Алгоритм Скала (с) |
|---|---|---|
| Большинство внутри | 0.4798681883 | 0.3421847262 |
| Большинство снаружи | 0.3491518315 | 0.2676950713 |
| Большинство на границе | 0.4048436849 | 0.2920872281 |

В таблице 3 приведены результаты измерения скорости для модифицированного в однородных координатах алгорима отсечения многоугольников и алгоритма Сазерленда—Ходгмана в зависимости от количества многоугольников.

Таблица 3 – Сравнение скорости работы алгоритмов в зависимости от объема входных данных

| Количество объектов | Алгоритм отсечения многоугольников в однородных координатах (с) | Алгоритм Сазерленда–Ходгмана (с) |
|---|---|---|
| 100 | 0.6035441705 | 0.3159855853 |
| 200 | 0.9915170796 | 0.6145738184 |
| 300 | 1.4992331124 | 0.9159397813 |
| 500 | 2.3123314878 | 1.5147858356 |
| 700 | 4.2208070337 | 2.1005305531 |
| 1000 | 6.4168028045 | 3.0366197334 |
| 5000 | 31.5481865403 | 15.9043921018 |
| 10000 | 63.8274154279 | 32.2366874475 |
| 20000 | 100.9731804108 | 67.4597061859 |
| 30000 | 198.7953923691 | 100.7255110244 |

В таблице 4 представлены результаты тестирования на скорость модифицированного в однородных координатах алгорима отсечения многоугольников и алгоритма Сазерленда–Ходгмана в зависимости от числа вершин отсекаемых многоугольников. Количество объектов в данном случае фиксированное.

Таблица 4 – Сравнение скорости работы алгоритмов в зависимости от вида многоугольников

| Вид многоугольников | Алгоритм отсечения многоугольников в однородных координатах (с) | Алгоритм Сазерленда–Ходгмана (с) |
|---|---|---|
| Треугольники | 0.6135387979 | 0.3685853246 |
| Четырехугольники | 0.7481679809 | 0.4781791854 |
| Пятиугольники | 1.0496690101 | 0.6088161338 |
| Шестиугольники | 1.1142174849 | 0.6519634172 |
| Семиугольники | 1.2605223425 | 0.7661390214 |
| Восьмиугольники | 1.4464939282 | 0.8644462691 |

В таблице 5 показаны результаты теста для модифицированного алгоритма отсечения многоугольников и стандартного алгоритма Сазерленда–Ходгмана в зависимости от положения объектов по отношению к области видимости. Количество объектов для каждого случая расположения неизменное.

Таблица 5 – Сравнение скорости работы алгоритмов в зависимости от разброса объектов относительно области видимости

| Расположение объектов | Алгоритм отсечения многоугольников в однородных координатах (с) | Алгоритм Сазерленда—Ходгмана (с) |
|---|---|---|
| Большинство внутри | 2.0872851702 | 1.3447997245 |
| Большинство снаружи | 0.8531331033 | 0.6277790892 |
| Большинство на границе | 1.6180335038 | 0.9269031311 |

Результаты, показанные в таблицах, демонстрируют, что в приведенной реализации время работы алгоритмов, оперирующих в однородных координатах, несколько больше оригинальных алгоритмов (Скала и Сазерленда—Ходгмана). Это может обуславливаться тем, что в качестве целого типа данных для координат был выбран тип `__int64`. Этот тип данных был выбран из-за того, что в результате операций векторного и скалярного произведения, регулярно использующихся в разработанных методах, отдельные координаты могут вырождаться в достаточно огромные величины. Использование целочисленных типов данных меньшей размерности приводило к некорректным результатам из-за переполнения.

# ЗАКЛЮЧЕНИЕ

Тематика, затронутая в проекте, актуальна, поскольку широко применяемые алгоритмы отсечения работают с вещественнозначными вычислениями, которые требуют дополнительных затрат памяти и времени. В связи с этим является целесообразным выполнение операций в целочисленных координатах.

В ходе дипломной работы были выполнены все поставленные задачи. Для проведения тестирования быстродействия алгоритмов отсечения отрезков и многоугольников были получены все необходимые входные данные, реализован тест скорости, что позволило произвести сравнение стандартных алгоритмов компьютерной графики и алгоритмов, модифицированных путем применения аппарата однородных координат.

Результаты показали, что в приведенной реализации алгоритмы, оперирующие в однородных координатах, работают медленнее стандартных. Это может обуславливаться тем, что они требуют операций с достаточно большими целыми числами, а это в некоторых случаях может привести к переполнению. При этом операции для работы с длинной арифметикой зачастую не являются более эффективными, чем операции с вещественными числами.

Настоящая тема заслуживает дальнейшего исследования, так как в целом отрицательные показатели для разработанных методов могут быть результатом выбора архитектуры для реализации. Желательно было бы повторить эксперименты на другой архитектуре для получения достоверного заключения.

Кроме того, стоит исследовать процедуры самих разработанных методов на предмет реорганизации операций, с целью уменьшения накопления величин и обход операций с длинной арифметикой.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1 *Мусхелишвили, Н. И.* Курс аналитической геометрии / Н. И. Мусхелишвили. — М.: МГУ, 1967.

2 Методические рекомендации по компьютерной графике [Электронный ресурс]. — URL: https://course.sgu.ru/pluginfile.php/54853/mod_resource/content/0/Lections/lections.pdf (Дата обращения 19.05.2017). Загл. с экр. Яз. рус.

3 Графика и обработка изображений [Электронный ресурс]. — URL: http://algolist.manual.ru/graphics/ (Дата обращения 17.05.2017). Загл. с экр. Яз. рус.

4 Отсечение многоугольников [Электронный ресурс]. — URL: http://compgraph.tpu.ru/Polygon.htm (Дата обращения 15.05.2017). Загл. с экр. Яз. рус.

5 Создание приложения Windows Forms в MVS [Электронный ресурс]. — URL: http://olocoder.ru/VS0.html (Дата обращения 17.05.2017). Загл. с экр. Яз. рус.


10.06.17