

# Алгоритмы определения коллизий аппроксимирующих цилиндров с моделью рельефа местности

Е.В. Страшнов, М.А. Торгашев

**Аннотация**—В статье рассматривается задача определения коллизий аппроксимирующих цилиндров, окружающих геометрию виртуальных объектов, с рельефом местности. Для решения этой задачи предлагается подход, в котором цилиндр задается аналитически, а рельеф представляется в виде регулярной прямоугольной полигональной сетки. В рамках предлагаемого подхода были разработаны алгоритмы, в которых пересечение определяется в зависимости от взаимного расположения цилиндра и плоскости элемента (прямоугольник или треугольник) полигональной сетки. Эти алгоритмы состоят из нескольких этапов и позволяют как определить сам факт пересечения, так и вычислить необходимую контактную информацию о пересечении. Предлагаемые решения базируются на быстрых геометрических тестах, активно используемых в компьютерной графике, что позволяет эффективно реализовать алгоритмы как на CPU, так и на GPU.

Разработанные алгоритмы определения коллизий аппроксимирующих цилиндров с рельефом местности были реализованы в виде программных модулей. Апробация этих модулей в созданном программном комплексе виртуального окружения показала адекватность и эффективность предлагаемых в статье решений.

**Ключевые слова**—определение коллизий, равномерная полигональная сетка, аппроксимирующий цилиндр, элемент, контактное многообразие, система виртуального окружения.

## I. ВВЕДЕНИЕ

В системах виртуального окружения и виртуальной реальности создается мир, в котором его объекты ведут себя аналогично объектам материальной реальности. При компьютерном моделировании виртуальные объекты могут перемещаться в виртуальном пространстве, что активно используется при создании компьютерной анимации и управлении игровым персонажем, а также в задачах, в которых осуществляется взаимодействие человека с виртуальными объектами. В процессе движения виртуальные объекты могут

сталкиваться друг с другом. Так как объекты виртуальные, то эти столкновения проявляются в пересечении (*коллизии*) этих объектов. Это приводит к необходимости решения задачи *определения коллизий* виртуальных объектов [1] между собой с последующим вычислением реакций (сил и моментов) на их столкновение (*для разрешения коллизий*). Соответствующие компьютерные программы (так называемые физические движки) обычно содержат вычислительные модули, в которых реализуется обработка пересечений виртуальных объектов. Распространенным подходом для определения коллизий является использование объектов стандартной формы. В рамках этого подхода виртуальные объекты окружаются набором аппроксимирующих контейнеров [2] (параллелепипеды, сферы, цилиндры и т.д.), и задача пересечения рассматривается относительно этих контейнеров. В данной статье рассматривается задача определения коллизий аппроксимирующих цилиндров с рельефом местности, заданным в виде *регулярной прямоугольной полигональной (состоящей из прямоугольников и треугольников) сетки*. Эта задача актуальна для моделирования в виртуальном окружении движения колес транспортных средств, качения бочек и перемещения других объектов, форма которых близка к цилиндрической. Важным требованием к таким алгоритмам является необходимость их работы в масштабе реального времени (т.е. все расчеты должны занимать не более 10 мсек).

Задаче определения коллизий виртуальных объектов в последние годы было посвящено множество работ. Для определения коллизий аппроксимирующих цилиндров с рельефом местности существует подход, в рамках которого поверхность рельефа задается набором выпуклых объектов, а рассматриваемая задача сводится к определению коллизий между этими объектами и цилиндром. Существующие решения для определения коллизий выпуклых объектов включают в себя алгоритмы, основанные на применении теоремы о разделяющей оси (см. [3], [4]) и семейство алгоритмов, которые носят название GJK-EPA (см. [5], [6], [7], [8]), для объектов, представленных неявно в виде выпуклой оболочки множества точек [9]. Недостатком такого подхода является большое число необходимых выпуклых объектов для задания рельефа, что не позволяет обеспечить масштаб реального времени.

Статья получена 29 мая 2020.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00387.

Е.В. Страшнов, ФГУ ФНЦ НИИСИ РАН (e-mail: strashnov\_evg@mail.ru).

М.А. Торгашев, кандидат физико-математических наук, ФГУ ФНЦ НИИСИ РАН (e-mail: mtorg@mail.ru).

При другом подходе рельеф местности представляется в виде полигональной сетки. Для такого подхода были предложены алгоритмы, в которых цилиндр также представляется в виде полигональной модели, а пересечение рассматривается между всеми треугольниками двух геометрий [10], [11]. Для обеспечения масштаба реального времени в [10] были задействованы полигональные модели со структурой данных в виде октодеревя, а в [11] расчет пересечений осуществлялся с применением стандарта OpenCL параллельных вычислений на GPU для задач общего назначения. Однако полигональная модель цилиндра содержит много треугольников, в то время как при аналитическом задании цилиндра достаточно указать лишь его высоту и радиус. С другой стороны, для произвольных полигональных сеток сложно правильно сформировать контактное множество. Поэтому также существует другой подход (см. [12], [13], [14], [15]), в котором виртуальные объекты задаются в аналити-

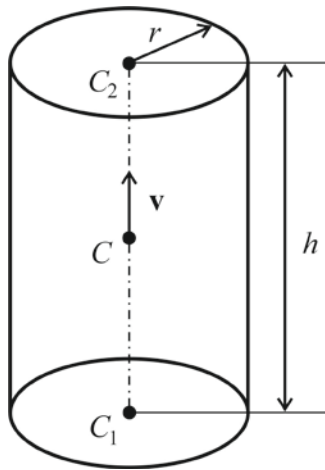


Рис. 1. Аппроксимирующий цилиндр

ческом виде. Однако, в этих работах задача определения коллизий аппроксимирующих цилиндров с рельефом местности не была решена в полном объеме.

В настоящей работе предлагаются новые методы и алгоритмы определения коллизий аппроксимирующих цилиндров, заданных в аналитическом виде, с рельефом местности. Для эффективной реализации алгоритмов предлагается подход, в котором рельеф представляется в виде регулярной прямоугольной полигональной сетки [16]. Преимущество такого подхода перед решениями, в которых применяется нерегулярная полигональная модель рельефа (например, [10], [11]), заключается в эффективном доступе к данным и независимости от детализации полигональной сетки. В предлагаемых алгоритмах используется ряд быстрых геометрических тестов на пересечение и отсечение отрезков в 2D, вычисление расстояний между примитивами и др., которые активно используются в компьютерной графике, что позволяет эффективно реализовать алгоритмы как на CPU, так и на GPU. При этом предполагается, что в процессе моделирования скорости виртуальных объектов сравнительно невелики, так что

за один шаг моделирования они не могут сильно проникнуть друг в друга. Предложенные в статье методы и алгоритмы были реализованы в созданном программном комплексе виртуального окружения и апробированы на примерах контакта объектов цилиндрической формы и колес робота с рельефом местности.

## II. ПОСТАНОВКА ЗАДАЧИ

Рассмотрим аппроксимирующий цилиндр радиуса  $r$  и высотой  $h$ , центр которого расположен в точке  $C$ , а ось имеет направление, заданное единичным вектором  $\mathbf{v}$  (все значения будем задавать в мировой системе координат, см. рис. 1). Основаниями этого цилиндра являются круги с центрами в точках  $C_1$  и  $C_2$ .

Регулярная прямоугольная сетка задается двумерным массивом [16], в котором записаны значения  $H_{ij}$  высот относительно некоторого нулевого уровня. Вершины сетки определяются равенствами

$$V_{ij} = (x_0 + jd_x, y_0 + id_y, H_{ij}), \quad j = 0, \dots, M, \quad i = 0, \dots, N, \quad (1)$$

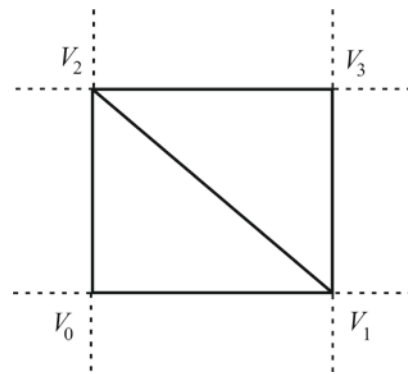


Рис. 2. Элемент полигональной сетки

где  $(x_0, y_0)$  – начальная точка прямоугольной сетки,  $d_x, d_y$  – шаги сетки вдоль осей  $x$  и  $y$ . Внутренней частью полигональной сетки мы будем считать область, которая располагается ниже нее.

С полигональной сеткой связано понятие элемента, которое мы в дальнейшем будем использовать. Для этого рассмотрим четыре смежные вершины  $V_0, V_1, V_2$  и  $V_3$  прямоугольной сетки (см. рис. 2), координаты которых вычисляются согласно (1):

$$V_0 = (x, y, H_0), \quad V_1 = (x + d_x, y, H_1),$$

$$V_2 = (x, y + d_y, H_2), \quad V_3 = (x + d_x, y + d_y, H_3).$$

Если  $\angle V_0 = 90^\circ$ , то

$$\mathbf{V}_0 \mathbf{V}_1 \cdot \mathbf{V}_0 \mathbf{V}_2 = (H_1 - H_0)(H_2 - H_0) = 0.$$

Отсюда следует, что либо  $H_1 = H_0$  либо  $H_2 = H_0$ .

Если вершины  $V_0, V_1, V_2$  и  $V_3$  находятся в одной плоскости и  $H_1 = H_0$  то  $H_2 = H_3$ . Тогда условие

$$\mathbf{V}_2 \mathbf{V}_3 \cdot \mathbf{V}_1 \mathbf{V}_3 = (H_3 - H_2)(H_3 - H_1) = 0,$$

при  $\mathbf{V}_2 \mathbf{V}_3 \neq \mathbf{0}$  и  $\mathbf{V}_1 \mathbf{V}_3 \neq \mathbf{0}$  означает, что  $\angle V_3 = 90^\circ$  и  $V_0 V_1 V_2 V_3$  является прямоугольником.

Аналогично можно показать, что  $V_0V_1V_2V_3$  будет прямоугольником, если  $H_2 = H_0$ .

Таким образом, если все вершины  $V_0, V_1, V_2$  и  $V_3$  лежат в одной плоскости и  $\angle V_0 = 90^\circ$ , то элементом является прямоугольник  $V_0V_1V_2V_3$ . В противном случае, элементами считаются треугольники  $V_0V_1V_2$  и  $V_3V_2V_1$ . В итоге получим, что для рассматриваемой полигональной модели рельеф местности образует связанное множество элементов, представляющих собой прямоугольники и треугольники.

Коллизия цилиндра и полигональной сетки характеризуется *контактным многообразием* (англ. *collision manifold*) – набором  $\Upsilon_{CP}$  троек параметров  $C_{P_i} = \{P_i, d_i, \mathbf{n}_i\}$ , где  $P_i$  –  $i$ -я точка контакта,  $\mathbf{n}_i$  – единичный вектор нормали в точке  $P_i$ , а  $d_i$  – глубина коллизии вдоль нормали  $\mathbf{n}_i$ . На рис. 3 приведен пример, в

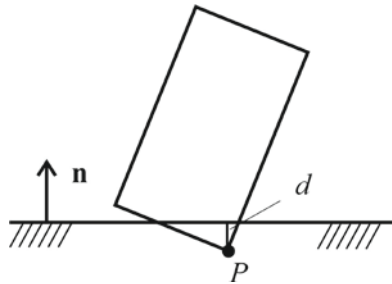


Рис. 3. Точка контакта, нормаль и глубина

котором пересечение цилиндра (вид сбоку) с плоскостью образует одну контактную точку с параметрами  $C_p = \{P, d, \mathbf{n}\}$ .

Таким образом, задача заключается в том, чтобы определить сам факт пересечения аппроксимирующего цилиндра с рельефом местности и, в случае пересечения, сформировать контактное многообразие  $\Upsilon_{CP}$ .

Решение этой задачи включает в себя три стадии: широкая фаза, узкая фаза и вычисление множества  $\Upsilon_{CP}$ . На стадии широкой фазы применяются грубые тесты на пересечение, чтобы отбросить те элементы полигональной сетки, с которыми цилиндр заведомо не пересекается. На стадии узкой фазы применяются более точные тесты, в которых определяется сам факт пересечения. И, наконец, на последней стадии вычисляется множество  $\Upsilon_{CP}$ . Опишем предлагаемые на этих стадиях алгоритмы более подробно.

### III. ШИРОКАЯ ФАЗА ОПРЕДЕЛЕНИЯ КОЛЛИЗИЙ

Для реализации широкой фазы определения коллизий аппроксимирующего цилиндра с рельефом местности предлагается подход, в рамках которого цилиндр окружается параллелепипедом AABV [1] (axis aligned bounding box – параллелепипед, выровненный по осям мировой системы координат). Покажем, как вычисляются границы AABV по оси  $x$ , определяющие отрезок  $[x_{\min}, x_{\max}]$ . Для этого спроецируем цилиндр на

плоскость  $Oxz$  в мировой системе координат (см. рис. 4).

Верхняя граница  $x_{\max}$  определяется крайней точкой  $N$  окружности основания цилиндра (при  $v_x \geq 0$  это будет верхнее основание цилиндра, как показано на рис. 4) и вычисляется следующим образом

$$x_{\max} = N_x = C_x + (\mathbf{CC}_2)_x + (\mathbf{C}_2\mathbf{N})_x. \quad (2)$$

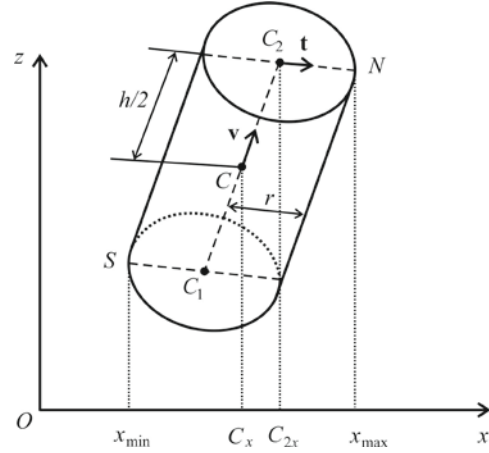


Рис. 4. Вычисление границ AABV по оси  $x$

Здесь

$$\mathbf{CC}_2 = \frac{h}{2} \mathbf{v} \text{ и } \mathbf{C}_2\mathbf{N} = r\mathbf{t}, \quad (3)$$

где  $\mathbf{t}$  – вектор, лежащий в верхнем основании цилиндра.

Для координаты  $t_x$  вектора  $\mathbf{t}$  можно показать, что справедливо равенство

$$|t_x| = \sqrt{1 - v_x^2}.$$

Используя это и подставляя (3) в (2), получим:

$$x_{\max} = C_x + \frac{h}{2}|v_x| + r\sqrt{1 - v_x^2}.$$

В силу симметрии нижняя граница  $x_{\min}$  вычисляется относительно крайней левой точки  $S$  как

$$x_{\min} = C_x - \frac{h}{2}|v_x| - r\sqrt{1 - v_x^2}.$$

Аналогичным образом выводятся выражения для границ AABV по осям  $y$  и  $z$ .

Прямоугольник, задаваемый границами  $[x_{\min}, x_{\max}]$  и  $[y_{\min}, y_{\max}]$ , используется для определения множества  $\Omega$  элементов, с которыми потенциально может пересекаться аппроксимирующий цилиндр (см. рис. 5). Это множество ограничено прямоугольником, индексы которого вычисляются следующим образом

$$i_{left} = \left\lfloor \frac{y_{\min} - y_0}{d_y} \right\rfloor; \quad i_{right} = \left\lceil \frac{y_{\max} - y_0}{d_y} \right\rceil; \quad (4)$$

$$j_{left} = \left\lfloor \frac{x_{\min} - x_0}{d_x} \right\rfloor; \quad j_{right} = \left\lceil \frac{x_{\max} - x_0}{d_x} \right\rceil,$$

где для левых границ используются округления снизу, а для правых – сверху. По этим индексам вычисляются вершины элементов согласно формуле (1).

Быстрым тестом является проверка пересечения относительно оси  $z$ . Для множества элементов  $\Omega$  вычисляется минимальное  $H_{\min}$  и максимальное  $H_{\max}$

значения высот рельефа. Если нижняя граница цилиндра  $z_{\min}$  больше, чем  $H_{\max}$ , то аппроксимирующий цилиндр и полигональная сетка не пересекаются. В противном

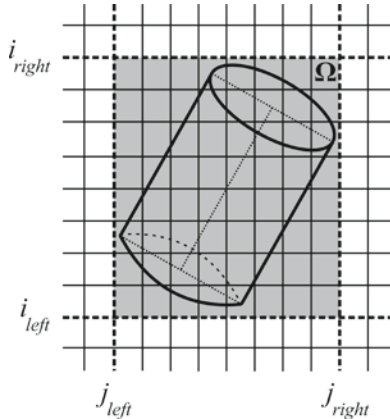


Рис. 5. Проекция цилиндра на горизонтальную плоскость

случае, еще дополнительно проверяется, является ли участок поверхности горизонтальной плоскостью. Это соответствует условию

$$|H_{\max} - H_{\min}| \leq \varepsilon_H,$$

где  $\varepsilon_H$  – задаваемая точность.

Если это условие выполнено, то проверяется пересечение аппроксимирующего цилиндра и плоскости. Уравнение плоскости имеет вид

$$s(P) = k_p - P \cdot \mathbf{n}_p = 0, \quad (5)$$

где  $P$  – точка, принадлежащая плоскости,  $\mathbf{n}_p$  – нормаль к плоскости,  $k_p$  – постоянная плоскости. Уравнение (5) применимо и для отдельного элемента. Например, если прямоугольник  $V_0V_1V_2V_3$  плоский, то параметры его плоскости вычисляются как

$$\mathbf{n}_p = \frac{\mathbf{V}_0\mathbf{V}_1 \times \mathbf{V}_0\mathbf{V}_2}{\|\mathbf{V}_0\mathbf{V}_1 \times \mathbf{V}_0\mathbf{V}_2\|}, \quad k_p = V_0 \cdot \mathbf{n}_p.$$

Для элемента, представляющего собой треугольник, эти параметры вычисляются аналогичным образом.

С алгоритмами определения коллизий аппроксимирующего цилиндра с плоскостью можно ознакомиться в работе [17].

Таким образом, широкую фазу определения коллизий аппроксимирующего цилиндра с полигональной сеткой можно записать в виде следующего Алгоритма:

1. Вычисляем границы AABB для цилиндра:

Если  $|v_z^2 - 1| \leq \varepsilon$ , то

$$x_{\min} = C_x - r; \quad x_{\max} = C_x + r; \\ y_{\min} = C_y - r; \quad y_{\max} = C_y + r; \quad z_{\min} = C_z - h/2,$$

иначе

$$\Delta x = (v_x^2 \leq 1) ? \sqrt{1 - v_x^2} : 0; \quad \Delta y = (v_y^2 \leq 1) ? \sqrt{1 - v_y^2} : 0;$$

$$\Delta z = (v_z^2 \leq 1) ? \sqrt{1 - v_z^2} : 0;$$

$$x_{\min} = C_x - \frac{h}{2}|v_x| - r\Delta x; \quad x_{\max} = C_x + \frac{h}{2}|v_x| + r\Delta x;$$

$$y_{\min} = C_y - \frac{h}{2}|v_y| - r\Delta y; \quad y_{\max} = C_y + \frac{h}{2}|v_y| + r\Delta y; \\ z_{\min} = C_z - \frac{h}{2}|v_z| - r\Delta z.$$

2. Вычисляем границы прямоугольной области по формулам (4) для множества элементов  $\Omega$ .
3. Вычисляем высоты  $H_{\min}$  и  $H_{\max}$  для множества  $\Omega$ .
4. Если выполнено  $H_{\max} < z_{\min}$ , то аппроксимирующий цилиндр и рельеф местности не пересекаются.
5. Если выполнено условие  $|H_{\max} - H_{\min}| \leq \varepsilon_H$ , то проверяем пересечение цилиндра с плоскостью.

*Алгоритм.* Широкая фаза определения коллизий.

Проверки в Алгоритме компонентов вектора  $\mathbf{v}$  при вычислении  $\Delta x$ ,  $\Delta y$  и  $\Delta z$  необходимы для обеспечения численной устойчивости к погрешностям вычислений. Если на выходе широкой фазы определено, что аппроксимирующий цилиндр пересекает участок рельефа местности, который не является горизонтальной плоскостью, то переходим к узкой стадии определения коллизий цилиндра с элементами из множества  $\Omega$ .

#### IV. УЗКАЯ ФАЗА ОПРЕДЕЛЕНИЯ КОЛЛИЗИЙ

Предлагаемый алгоритм узкой стадии определения коллизий аппроксимирующего цилиндра с элементом обрабатывает три возможных случая контакта цилиндра с элементом: а) пересечение элемента с основанием цилиндра; б) пересечение с боковой поверхностью и в) имеется только одна точка контакта элемента с основанием цилиндра. Каждый из этих случаев соответствует проверке расположения точек оснований цилиндра

$$A_i = C_i + r\mathbf{w}, \quad B_i = C_i - r\mathbf{w}, \quad i = 1, 2$$

относительно плоскости элемента, где единичный вектор  $\mathbf{w}$  вычисляется как

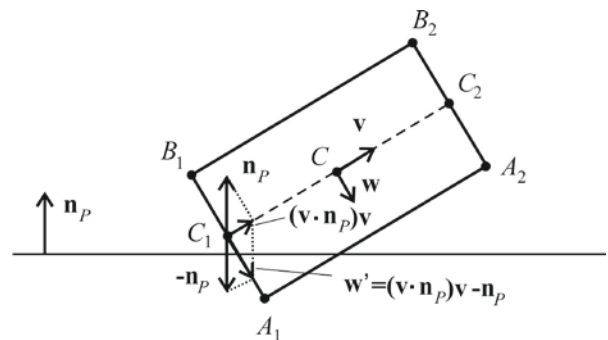


Рис. 6. Взаимное расположение цилиндра и плоскости

$$\mathbf{w} = \begin{cases} \frac{\mathbf{V}_0\mathbf{V}_1}{\|\mathbf{V}_0\mathbf{V}_1\|}, & \text{если } \mathbf{v} \perp \mathbf{n}_p; \\ \frac{(\mathbf{v} \cdot \mathbf{n}_p)\mathbf{v} - \mathbf{n}_p}{\|(\mathbf{v} \cdot \mathbf{n}_p)\mathbf{v} - \mathbf{n}_p\|}, & \text{в прот. сл.} \end{cases}$$

где  $V_0$  и  $V_1$  – вершины элемента. Вектор  $\mathbf{w}$  построен таким образом, чтобы были выполнены условия

$\mathbf{w} \cdot \mathbf{v} = 0$  и  $\mathbf{w} \cdot \mathbf{n}_p \leq 0$ .

На рис. 6 введенные точки и вектор  $\mathbf{w}$  схематично показаны для случая, когда имеется только одна точка контакта.

Проверка расположения точек относительно плоскости осуществляется с помощью уравнения (5). Если точка  $P$  основания цилиндра лежит выше плоскости элемента, то  $s(P) < 0$ , в противном случае  $s(P) \geq 0$ . На рис. 6 для точки  $A_1$  будет  $s(A_1) \geq 0$ , а для точки  $A_2$  –  $s(A_2) < 0$ .

Для каждого случая взаимного расположения цилиндра и плоскости элемента требуется определить сам факт пересечения и добавить, если необходимо, точки контакта в множество  $L(C_p)$  потенциальных точек.

Если контактные точки находятся в одной плоскости, то это множество далее будем называть *пятном контакта*. Введем некоторые вспомогательные множества вершин и элементов. В множество  $G_F$  будем записывать вершины контактирующих элементов, а в множестве  $G_I$  будем накапливать вершины элементов, находящихся внутри пятна контакта. При этом сами эти элементы образуют множество  $\Omega_E$ , а элементы, которые полностью находятся внутри пятна контакта, образуют множество  $\Omega_I$ . Обработка элементов состоит из двух этапов. Рассмотрим первый этап.

#### A. Контакт нижнего основания цилиндра с элементом

В случае контакта нижнего основания цилиндра с элементом должно быть выполнено условие

$$s(A_1) \geq 0, s(A_2) < 0 \text{ и } s(B_1) \geq 0, \quad (6)$$

Это означает, что обе точки  $A_1$  и  $B_1$  нижнего основания лежат под плоскостью элемента (или касаются ее), а нижняя точка верхнего основания лежит выше этой плоскости. Для проверки пересечения цилиндра с элементом спроецируем нижнее основание цилиндра на плоскость. В общем случае при проецировании нижнее основание переходит в эллипс (см. рис. 7), а центр  $C_1$  основания – в точку  $\tilde{C}_1 = C_1 + s(C_1)\mathbf{n}_p$ , где  $s(C_1)$  вычисляется по формуле (5). Аналогичным образом вычисляются проекции  $\tilde{A}_1$  и  $\tilde{D}_1$  двух точек  $A_1$  и  $D_1$  основания цилиндра, где  $D_1 = C_1 + r\mathbf{u}$ ,  $\mathbf{u} = \mathbf{v} \times \mathbf{w}$  – единичный вектор в плоскости нижнего основания, перпендикулярный вектору  $\mathbf{w}$ .

В системе координат  $\tilde{C}_1 x_1 y_1$  уравнение эллипса записывается в виде

$$E(x_1, y_1) = 1 - \frac{x_1^2}{a^2} - \frac{y_1^2}{b^2} = 0, \quad (7)$$

где  $a = |\tilde{C}_1 \tilde{A}_1|$ ,  $b = |\tilde{C}_1 \tilde{D}_1|$  – полуоси эллипса.

Для проверки пересечения элемента с эллипсом вычисляется точка элемента, которая является ближайшей к центру  $\tilde{C}_1$  эллипса. На рис. 7 для прямоу-

гольника  $V_0 V_1 V_2 V_3$  ближайшей является точка  $Q$ , а для треугольника  $V_0 V_1 V_2$  – точка  $V_1$ . Как вычислять эту точку для прямоугольника и треугольника подробно описано в работе [16]. Ее компоненты  $\tilde{q}_x$  и  $\tilde{q}_y$  в системе координат  $\tilde{C}_1 x_1 y_1$  вычисляются следующим образом

$$\tilde{q}_x = \tilde{C}_1 Q \cdot \xi, \quad \tilde{q}_y = \tilde{C}_1 Q \cdot \eta,$$

где  $\xi$  и  $\eta$  – единичные векторы осей эллипса.

Если для полученных координат выполнено

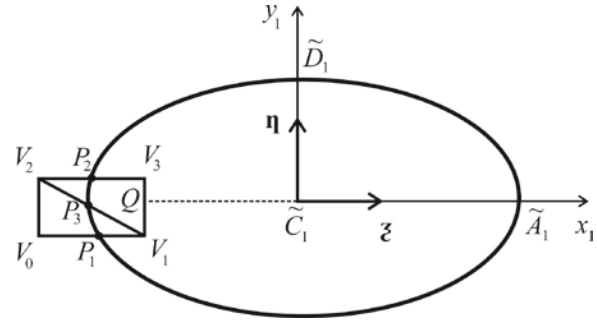


Рис. 7. Пересечение эллипса и элемента

неравенство  $E(\tilde{q}_x, \tilde{q}_y) < 0$ , то цилиндр и элемент не пересекаются.

Если элемент пересекает эллипс, то точки пересечения являются потенциальными точками контактами и добавляются в множество  $L(C_p)$ . Точка пересечения вычисляется на основе параметрического задания отрезка  $P(t) = V_i + t(\mathbf{V}_i - \mathbf{V}_j)$  в системе координат  $\tilde{C}_1 x_1 y_1$ , где  $t \in [0, 1]$  – параметр отрезка,  $V_i$  и  $V_j$  – вершины элемента, которые находятся внутри и снаружи эллипса, соответственно. Подставляя координаты точки в (7), получим квадратное уравнение относительно параметра  $t$ :

$$k_0 t^2 + 2k_1 t + k_2 = 0,$$

где  $k_0 = \frac{(\tilde{V}_{j,x} - \tilde{V}_{i,x})^2}{a^2} + \frac{(\tilde{V}_{j,y} - \tilde{V}_{i,y})^2}{b^2}$ ,

$$k_1 = \frac{\tilde{V}_{i,x}(\tilde{V}_{j,x} - \tilde{V}_{i,x})}{a^2} + \frac{\tilde{V}_{i,y}(\tilde{V}_{j,y} - \tilde{V}_{i,y})}{b^2}, \quad k_2 = \frac{\tilde{V}_{i,x}^2}{a^2} + \frac{\tilde{V}_{i,y}^2}{b^2} - 1.$$

Решением квадратного уравнения является параметр  $t_p \in [0, 1]$ , по которому определяются координаты точки  $P$  в системе координат  $\tilde{C}_1 x_1 y_1$ :

$$\tilde{P}_x = \tilde{V}_{i,x} + t_p(\tilde{V}_{j,x} - \tilde{V}_{i,x}), \quad \tilde{P}_y = \tilde{V}_{i,y} + t_p(\tilde{V}_{j,y} - \tilde{V}_{i,y}).$$

Контактная точка вычисляется путем обратного проецирования на окружность основания цилиндра. Для полученной точки вычисляется нормаль (совпадающая с нормалью к плоскости элемента) и глубина проникновения вдоль этой нормали. В итоге в множество контактных точек  $L(C_p)$  добавляется точка

$C_p = \{P, d, \mathbf{n}\}$  со следующими параметрами:

$$P = C_1 + \frac{\tilde{P}_x}{a} \mathbf{C}_1 \mathbf{A}_1 + \frac{\tilde{P}_y}{b} \mathbf{C}_1 \mathbf{D}_1, \quad d = s(P), \quad \mathbf{n} = \mathbf{n}_p. \quad (8)$$

Все вершины элемента, пересекающего эллипс, добавляются во множество  $G_F$ . Те вершины элемента, которые находятся внутри эллипса, также добавляются во множество  $G_I$ .

Если элемент полностью находится внутри эллипса, то элемент добавляется во множество  $\Omega_I$  с сортировкой по возрастанию расстояний до границы эллипса, вычисляемых с помощью уравнения (7).

*В. Контакт боковой поверхности цилиндра с*

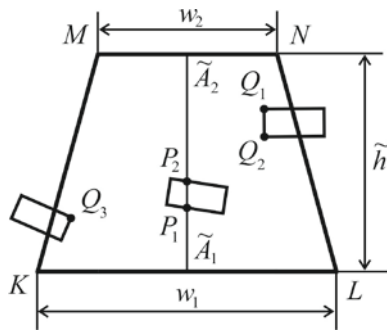


Рис. 8. Пересечение трапеции и элемента

элементом

При контакте боковой поверхности цилиндра с элементом должны выполняться условия

$$s(A_1) \geq 0 \text{ и } s(A_2) \geq 0. \quad (9)$$

Пересечение цилиндра с плоскостью элемента в общем случае образует трапецию  $KLMN$  (см. рис. 8), стороны и высота которой вычисляются как

$$w_i = \sqrt{s(A_i)(2r - s(A_i))}, \quad \tilde{h} = |\tilde{A}_1\tilde{A}_2|, \quad i = 1, 2,$$

где  $\tilde{A}_i$  – проекции точек  $A_i$  на плоскость элемента.

Для проверки пересечения трапеции с прямоугольником (или треугольником) в работе был задействован алгоритм на основе SAT [3] (Separating Axis Theorem – теорема о разделяющей оси). Идея алгоритма заключается в том, что если существует такая ось, что проекции двух фигур на нее не пересекаются, то и сами фигуры не пересекаются. Такая ось называется разделяющей. Если такой оси нет, то рассматриваемые фигуры пересекаются. При этом, для проверки

пересечения двух многоугольников достаточно рассмотреть только оси, направления которых перпендикулярны сторонам многоугольников. Если элемент прямоугольный, то мы имеем три оси, перпендикулярные сторонам трапеции и две оси, перпендикулярные сторонам прямоугольника (всего пять осей). Если ни одна из пяти осей не является разделяющей, то проекция цилиндра пересекается с элементом. Если элемент пересекает проекцию (отрезок  $\tilde{A}_1\tilde{A}_2$ ) оси цилиндра, то граничные точки пересечения считаются потенциальными точками контакта (на рис. 8 это точки  $P_1$  и  $P_2$ ). В противном случае, точки, которые являются ближайшими к отрезку  $\tilde{A}_1\tilde{A}_2$  (на рис. 8 это точки  $Q_1$ ,  $Q_2$  и  $Q_3$ ), могут быть контактными точками в том случае, если они не принадлежат элементу, который был уже задействован в формировании контактной точки. Такая ситуация возможна только, если в конце первого этапа не был найден хотя бы один элемент, который пересекает отрезок  $\tilde{A}_1\tilde{A}_2$ .

Для тех элементов, которые пересекают отрезок  $\tilde{A}_1\tilde{A}_2$ , вычисляются крайние точки пересечения. Для этого в данной работе был задействован алгоритм Кируса-Бека [18], предназначенный для отсекаания отрезков произвольным выпуклым многоугольником. Здесь отрезком является  $\tilde{A}_1\tilde{A}_2$ , а многоугольником является элемент. Выходом алгоритма являются параметры  $t_1$  и  $t_2$  отсекания элементом отрезка  $\tilde{A}_1\tilde{A}_2$ . Если  $t_i \in [0, 1]$ , то элемент пересекает отрезок, а концы отсеченного отрезка используются для формирования контактных точек  $C_{p,i} = \{P_i, d_i, \mathbf{n}_i\}$  с параметрами

$$P_i = A_1 + t_i \mathbf{A}_1 \mathbf{A}_2, \quad d_i = s(P_i), \quad \mathbf{n}_i = \mathbf{n}_p, \quad i = 1, 2, \quad (10)$$

которые добавляются в множество  $L(C_p)$ . Вершины этого элемента добавляются во множество  $G_F$ .

Для остальных элементов вычисляются ближайшие точки  $Q_i$ ,  $i = 1, 2$  и расстояние  $d_{\min}$  до отрезка  $\tilde{A}_1\tilde{A}_2$ . Эти элементы добавляются во множество  $\Omega_E$  с сортировкой расстояний по возрастанию.

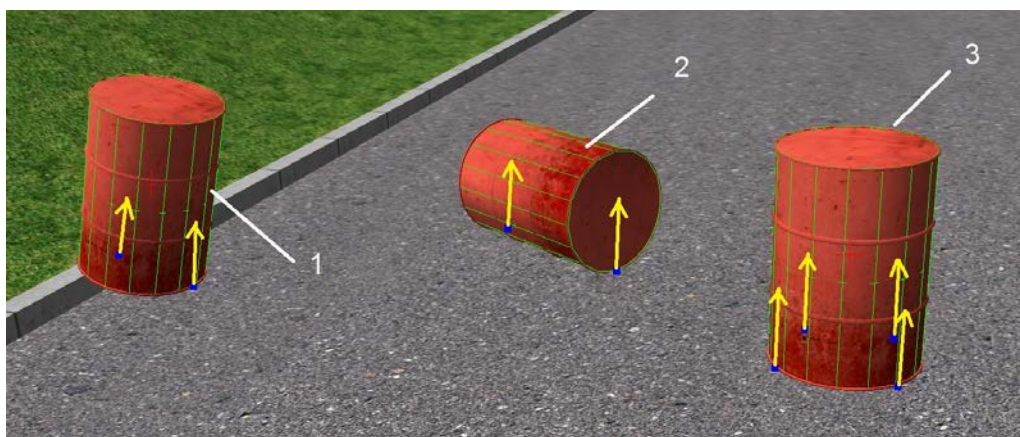


Рис. 9. Контакт бочек с асфальтом в виртуальной сцене

### С. Контакт цилиндра с элементом в одной точке

В случае контакта нижнего основания цилиндра с элементом в одной точке должно быть выполнено условие

$$s(A_1) \geq 0, s(A_2) < 0 \text{ и } s(B_1) < 0. \quad (11)$$

Для определения коллизии в этом случае вычисляется точка  $Q$  [16], которая является ближайшей к точке  $\tilde{A}_1 = A_1 - s(A_1)\mathbf{n}_p$  (проекция  $A_1$  на плоскость элемента). Проверка на пересечение цилиндра с элементом состоит в том, чтобы определить, находится ли точка  $Q$  внутри цилиндра. Для этого вычисляются координата  $z_Q$  в локальной системе координат цилиндра и расстояние  $r_Q$  до оси цилиндра. Если  $|z_Q| > h/2$  или  $r_Q > r$ , то цилиндр и элемент не пересекаются.

Если точка  $Q$  совпадает с точкой  $\tilde{A}_1$ , то контактная точка  $C_p = \{P, d, \mathbf{n}\}$  с параметрами

$$P = A_1, d = s(A_1), \mathbf{n} = \mathbf{n}_p, \quad (12)$$

добавляется в множество  $L(C_p)$ . Вершины этого элемента добавляются во множество  $G_F$ .

Если точка  $Q$  не совпадает с точкой  $\tilde{A}_1$ , то элемент добавляется во множество  $\Omega_E$  с сортировкой расстояний  $d_{\min} = |Q\tilde{A}_1|$  до точки  $\tilde{A}_1$ .

### Д. Второй этап обработки элементов

На втором этапе алгоритма обрабатываются множества элементов  $\Omega_I$  и  $\Omega_E$ , полученные на первом этапе. Множество  $\Omega_I$  сформировано для случая коллизии основания цилиндра с элементами и состоит из тех элементов, которые целиком находятся внутри пятна контакта. Это множество является упорядоченным по расстояниям от ближайшей точки элемента до проекции окружности основания цилиндра на плоскость элемента. Задача обработки множества  $\Omega_I$  заключается в том, чтобы определить, принадлежат ли вершины

элементов множества  $\Omega_I$  границе пятна контакта. Если вершина элемента из множества  $\Omega_I$  не принадлежит множеству  $G_F$ , то эта вершина добавляется во множества  $G_I$  и  $G_F$ . В противном случае, вершина удаляется из множества  $G_I$ , так как она гарантированно находится внутри пятна контакта. Если после обработки всех элементов из множества  $\Omega_I$  во множестве  $G_I$  останутся вершины, то эти вершины находятся на границе пятна контакта и их требуется добавить в множество  $L(C_p)$  с формированием контактной точки по формуле, которая аналогична (8).

Множество  $\Omega_E$  сформировано из элементов, полученных при рассмотрении коллизии боковой поверхности цилиндра с элементами, включая случай, когда есть только одна точка контакта. Это множество тоже является упорядоченным по расстояниям от ближайших точек элемента до проекций отрезка боковой поверхности цилиндра или точки на плоскость элемента. Если ближайшие точки  $Q_i$ ,  $i=1,2$  элемента не находятся на ребрах, соединяющих вершины  $G_F$ , то эти точки принадлежат другому пятну контакта. Для этих точек вычисляются координата  $z_{Q,i}$  в локальной системе координат цилиндра и расстояние  $r_{Q,i}$  до оси цилиндра. Затем вычисляется ближайшая точка  $Q'_i = C + z_{Q,i}\mathbf{v}$  оси цилиндра и определяется, какая часть цилиндра является ближайшей к точке  $Q_i$  путем вычисления расстояний

$$l_{i,1} = h/2 - |z_{Q,i}|, l_{i,2} = r - r_{Q,i}, i=1,2.$$

Если  $l_{i,1} < l_{i,2}$  и ось цилиндра не параллельна плоскости элемента, что соответствует условию  $|\mathbf{v} \cdot \mathbf{n}_p| \geq \varepsilon_v$ , то точка контакта  $C_{p,i} = \{P_i, d_i, \mathbf{n}_i\}$  находится на основании цилиндра и имеет параметры:

$$P_i = Q_i + \text{sgn}(z_{Q,i})l_{i,1}\mathbf{v}, d_i = l_{i,1}, \mathbf{n}_i = -\text{sgn}(z_{Q,i})\mathbf{v}. \quad (13)$$

В противном случае, точка контакта  $C_{p,i} = \{P_i, d_i, \mathbf{n}_i\}$



Рис. 10. Движение модели колесного робота КРТ-100МП по сложному рельефу

находится на боковой поверхности и имеет параметры:

$$P_i = Q_i' - r\mathbf{n}_i, d_i = l_{i,2}, \mathbf{n}_i = \mathbf{Q}_i\mathbf{Q}_i' / r_{Q_i}. \quad (14)$$

Полученная точка контакта  $C_{p,i}$  добавляется в  $L(C_p)$ .

В конце обработки все вершины элемента добавляются в множество  $G_F$ .

## V. ПОСТРОЕНИЕ КОНТАКТНОГО МНОГООБРАЗИЯ

На этой стадии для полученного множества  $L(C_p)$  требуется построить контактное многообразие  $\Upsilon_{CP}$  из тех контактных точек, которые будут использоваться на стадии разрешения коллизии. Для этого разобьем  $L(C_p)$  на подмножества  $L_k(C_p)$ ,  $k=1, \dots, K$  тех контактных точек, которые имеют одинаковые нормали (здесь  $K$  – число подмножеств). Все точки  $P_i^k$  из  $L_k(C_p)$  лежат в одной плоскости и будут образовывать отдельное пятно контакта.

Для построения контактного многообразия  $\Upsilon_{CP}$  в масштабе реального времени в данной работе предлагается следующий эвристический алгоритм. Если количество контактных точек в  $L_k(C_p)$  больше четырех, то из  $L_k(C_p)$  выбираются четыре точки, которые образуют четырехугольник с максимальной площадью. На первом шаге выбирается точка  $C_{p,1}^k$ , которая имеет максимальную глубину проникновения  $d_{\max}$ . На втором шаге выбирается точка  $C_{p,2}^k$ , для которой  $P_2^k$  наиболее удалена от  $P_1^k$ . Затем на третьем шаге выбирается точка  $C_{p,3}^k$ , для которой  $P_3^k$  наиболее удалена от отрезка  $P_1^k P_2^k$ . Наконец, на четвертом шаге из оставшихся точек из множества  $L_k(C_p)$  выбирается точка  $C_{p,4}^k$ , для которой площадь четырехугольника  $P_1^k P_2^k P_3^k P_4^k$  будет максимальна.

Если количество контактных точек меньше или равно четырем, то все точки отбираются. Из полученных точек  $C_{p,i}^k$  образуется контактное многообразие  $\Upsilon_{CP}$ .

Алгоритм также учитывает частный случай, характерный для случая контакта боковой поверхности с рельефом местности, когда все контактные точки лежат на одной прямой. В этом случае, если после вычисления третьей точки выполнено условие

$$\|\mathbf{P}_1^k \mathbf{P}_2^k \times \mathbf{P}_2^k \mathbf{P}_3^k\| \leq \varepsilon_2,$$

то сохраняются только две наиболее удаленные точки.

## VI. ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена задача определения коллизий аппроксимирующих цилиндров с рельефом местности, заданным в виде регулярной прямоугольной полигональной сетки. Предложенное решение включает новые алгоритмы определения пересечений и построения контактного многообразия, основанные на эффективных геометрических тестах. Апробация разработанных в статье алгоритмов определения

коллизий аппроксимирующих цилиндров с рельефом местности была проведена на виртуальном полигоне в программном комплексе виртуального окружения [19], [20], созданном в ФГУ ФНЦ НИИСИ РАН. Этот программный комплекс включает в себя подсистемы управления, динамики и визуализации. Апробация проводилась на двух примерах. В первом примере (см. рис. 9) рассматривался контакт цилиндрической бочки с асфальтом и бордюром (бочка № 1), контакт боковой поверхности (бочка № 2) и контакт основания (бочка № 3). На рис. 9 синими точками показаны точки контакта, а желтыми стрелками – их нормали. Во втором примере (см. рис. 10) рассматривалось моделирование движения виртуальной модели колесного робота КРТ-100МП по сложному рельефу местности. Данный робот состоит из основного корпуса и тележки с грузом. При моделировании рассматривалась ситуация, когда корпус робота своей передней частью наезжает на холм из песка, а тележка находится на травяном грунте.

Результаты апробации в комплексе подтвердили эффективность и адекватность разработанных алгоритмов определения коллизий аппроксимирующих цилиндров с рельефом местности для решения задач в системах виртуального окружения, имитационно-тренажерных комплексах, образовательных приложений и т.д.

## БИБЛИОГРАФИЯ

- [1] C. Eicson, *Real-time collision detection*. San Francisco: Morgan Kaufmann Publishers, 2005.
- [2] Bounding volume. Available: [https://en.wikipedia.org/wiki/Bounding\\_volume](https://en.wikipedia.org/wiki/Bounding_volume).
- [3] J. Huynh, "Separating axis theorem for oriented bounding boxes," 2009, Available: <http://www.jkh.me/files/tutorials/Separating%20Axis%20Theorem%20for%20Oriented%20Bounding%20Boxes.pdf>.
- [4] D. Gregorius, "The separating axis test," *In GDC*, 2013.
- [5] L. OIvang, "Real-time collision detection with implicit objects," Department of Information Technology, Uppsala University, Sweden, 2010.
- [6] А.В. Санников, "Численное моделирование динамики систем твердых деформируемых и жестких тел," Диссертация на соискание ученой степени кандидата физ.-мат. наук, Москва, 2015.
- [7] G. van den Bergen, and D. Gregorius, *Game physics pearls*, AK Peters/CRC Press, 2010.
- [8] J.A. Newth, "Minkowski portal refinement and speculative contacts in Box2D," Master's Projects, 2013.
- [9] D. Gregorius, "Implementing quickhull," *Game Developers Conference (Valve Software) in San Francisco*, 2014.
- [10] F.J. Melero, A. Aguilera, and F.R. Feito, "Fast collision detection between high resolution polygonal models," *Computers & Graphics*, 2019, pp. 97-106.
- [11] N. Backman, "Collision detection of triangle meshes using GPU," Master's thesis, Umea University, 2010.
- [12] A. Moravanzky, and P. Terdiman, "Fast Contact Reduction for Dynamics Simulation," *In Game Programming Gems 4*, Charles River Media, 2004, pp. 253-263.
- [13] P. Terdiman, "Contact generation for meshes", Available: <http://www.codercorner.com/MeshContacts.pdf>.
- [14] E.-A. Karabassi, G. Papaioannou, T. Theoharis, and A. Boehm, "Intersection test for collision detection in particle systems," *Journal of Graphic Tools*, 1999, pp. 25-37.
- [15] M. G. Coutinho, *Dynamic simulation of multibody systems*. Springer, 2001.
- [16] Е.В. Страшнов, Л.А. Финагин, "Определение коллизий аппроксимирующих сфер с рельефом местности," *Труды НИИСИ РАН*, Т. 9, № 5, 2019, стр. 111-118.



- [17] Е.В. Страшнов, Д.В. Омельченко, "Определение коллизий аппроксимирующих цилиндров со сферами и плоскостью," *Труды НИИСИ РАН*, Т. 8, № 2, 2018, стр. 92-97.
- [18] M. Szygus, and J. Beck. "Generalized two- and three-dimensional clipping," *Computers & Graphics*, 1978, pp. 23–28.
- [19] Е.В. Страшнов, М.В. Михайлюк, "Методы силового управления манипуляционными роботами в системах виртуального окружения," *International Journal of Open Information Technologies*, Т. 7, № 9, 2019, стр. 39-45.
- [20] П.Ю. Тимохин, М.В. Михайлюк, А.В. Мальцев, "Построение на GPU в масштабе реального времени адаптивной модели рельефа Земли на основе эллипсоида вращения," *International Journal of Open Information Technologies*, Т. 7, № 10, 2019, стр. 22-35.

# Collision detection algorithms of bounding cylinders with terrain model

E.V. Strashnov, M.A. Torgashev

**Abstract**—This paper considers the task of collision detection for bounding cylinders surrounding the geometry of virtual objects with a terrain. To solve this task, an approach is proposed in which the cylinder is defined analytically, and the terrain is represented as a regular rectangular polygonal mesh. As part of the proposed approach, algorithms have been developed in which the intersection is determined depending on the relative position of the cylinder and the plane of the polygonal mesh element (rectangle or triangle). These algorithms consist of several stages and allow to detect both the fact of the intersection and calculate the necessary contact information about intersection. The proposed solutions are based on fast geometric tests, which are actively used in computer graphics, and that allows to effectively implement algorithms on both CPU and GPU.

The developed algorithms for collision detection of bounding cylinders with terrain were implemented in the form of software modules. Approbation of these modules in the created software package of the virtual environment showed the adequacy and effectiveness of the proposed solutions in this paper.

**Keywords**—collision detection, regular polygonal mesh, bounding cylinder, regular height grid, element, contact manifold, virtual environment system.

## REFERENCES

- [1] C. Eicson, Real-time collision detection. San Francisco: Morgan Kaufmann Publishers, 2005.
- [2] Bounding volume. Available: [https://en.wikipedia.org/wiki/Bounding\\_volume](https://en.wikipedia.org/wiki/Bounding_volume).
- [3] J. Huynh, "Separating axis theorem for oriented bounding boxes," 2009, Available: <http://www.jkh.me/files/tutorials/Separating%20Axis%20Theorem%20for%20Oriented%20Bounding%20Boxes.pdf>.
- [4] D. Gregorius, "The separating axis test," In GDC, 2013.
- [5] L. Olvang, "Real-time collision detection with implicit objects," Department of Information Technology, Uppsala University, Sweden, 2010.
- [6] A. V. Sannikov, "Numerical simulation of rigid, deformable and stiff bodies," Ph.D. dissertation, Moscow, 2015.
- [7] G. van den Bergen, and D. Gregorius, Game physics pearls, AK Peters/CRC Press, 2010.
- [8] J.A. Newth, "Minkowski portal refinement and speculative contacts in Box2D," Master's Projects, 2013.
- [9] D. Gregorius, "Implementing quickhull," Game Developers Conference (Valve Software) in San Francisco, 2014.
- [10] F.J. Melero, A. Aguilera, and F.R. Feito, "Fast collision detection between high resolution polygonal models," Computers & Graphics, 2019, pp. 97-106.
- [11] N. Backman, "Collision detection of triangle meshes using GPU," Master's thesis, Umea University, 2010.
- [12] A. Moravanszky, and P. Terdiman, "Fast Contact Reduction for Dynamics Simulation," In Game Programming Gems 4, Charles River Media, 2004, pp. 253–263.
- [13] P. Terdiman, "Contact generation for meshes", Available: <http://www.codercorner.com/MeshContacts.pdf>.
- [14] E.-A. Karabassi, G. Papaioannou, T. Theoharis, and A. Boehm, "Intersection test for collision detection in particle systems," Journal of Graphic Tools, 1999, pp. 25-37.
- [15] M. G. Coutinho, Dynamic simulation of multibody systems. Springer, 2001
- [16] E.V. Strashnov, and L.A. Finagin, "Collision detection of bounding spheres with terrain," Proceedings of NIISI RAS, Vol. 9, No. 5, 2019, pp. 111-118.
- [17] E.V. Strashnov, and D.V. Omelchenko, "Collision detection of bounding cylinders with spheres and a plane," Proceedings of NIISI RAS, Vol. 8, No. 2, 2018, pp. 92-97.
- [18] M. Cyrus, and J. Beck. "Generalized two- and three-dimensional clipping," Computers & Graphics, 1978, pp. 23–28.
- [19] E.V. Strashnov, and M.V. Mikhaylyuk, "Methods for robotic manipulator force control in virtual environment systems," International Journal of Open Information Technologies, Vol. 7, No. 9, 2019, pp. 39-45.
- [20] P. Yu. Timokhin, M.V. Mikhaylyuk, and A.V. Maltsev, "Real-time construction on the GPU of adaptive terrestrial relief model based on the spheroid," International Journal of Open Information Technologies, Vol. 7, No. 10, 2019, pp. 22-35.