

ПРЕДСТАВЛЕНИЕ ТРЁХМЕРНЫХ ОБЪЕКТОВ ДЛЯ ЛОГИЧЕСКИХ ПРЕОБРАЗОВАНИЙ В РЕАЛЬНОМ ВРЕМЕНИ

И. Е. Плотников¹ [0009-0004-2891-9913], Д. И. Костюк² [0000-0002-6542-6980]

^{1, 2}Казанский (Приволжский) федеральный университет, г. Казань, 420008, Россия

¹ilaya.link@gmail.com, ²xdxnkx@gmail.com

Аннотация

Дан сравнительный анализ методов представления трёхмерных объектов для выполнения булевых операций в реальном времени в среде игрового движка Unity. Рассмотрены четыре основных подхода: полигональное представление на основе конструктивной твердотельной геометрии (CSG), функции знакового расстояния (SDF), воксельные методы и CAD-системы с представлением границ (B-Rep) и NURBS-поверхностями.

Проведено экспериментальное исследование производительности полигональных алгоритмов булевых операций и SDF-функций на основе реализации ray marching. Выявлено, что полигональные методы характеризуются высокими начальными затратами на построение системы, но обеспечивают стабильную производительность при длительных операциях и сохранение результатов преобразований. SDF-функции демонстрируют высокую скорость выполнения операций и гибкость в создании сглаженных переходов между объектами, однако ограничены в применении для долговременных задач из-за особенностей вычислительной модели.

Определены области эффективного применения каждого подхода: полигональные методы рекомендуются для задач, требующих точного геометрического контроля и интеграции с традиционными графическими конвейерами, в то время как SDF-функции оптимальны для процедурной генерации, многослойного рендеринга материалов и создания динамических визуальных эффектов. Результаты исследования могут быть использованы при разработке интерактивных симуляторов, игровых приложений и систем виртуальной реальности.

Ключевые слова: булевы операции, трёхмерное моделирование, конструктивная твердотельная геометрия, функции знакового расстояния, SDF-функции, Unity, реальное время, полигональные сетки, ray marching.

ВВЕДЕНИЕ

Булевы операции являются важным инструментом в трёхмерном моделировании, позволяя формировать новые геометрические объекты путём логического комбинирования базовых форм (см. рис. 1). Теоретические основы булевых операций над твердотельными объектами были заложены в [1], где впервые систематизированы принципы представления и обработки трёхмерной геометрии. Применение таких операций востребовано в различных сферах, включая проектирование изделий, архитектурное моделирование, создание интерактивного и игрового контента, а также в системах виртуальной и дополненной реальности.

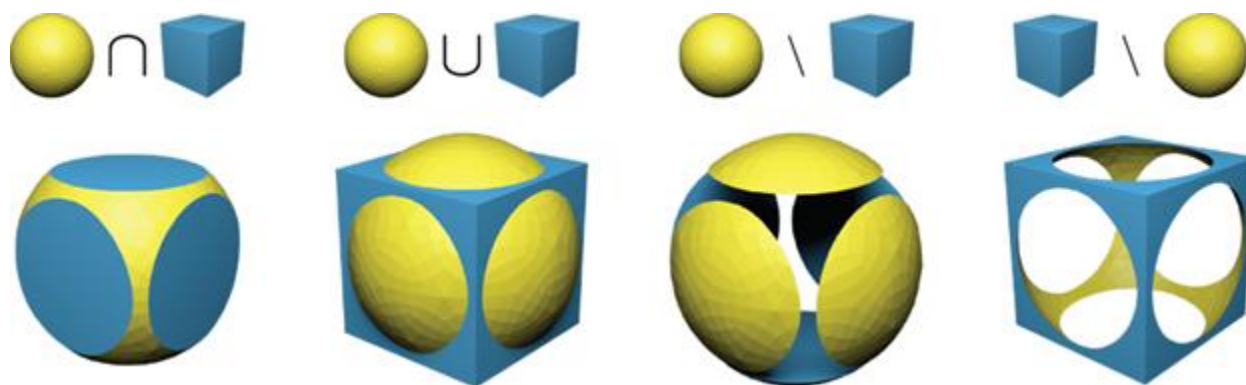


Рис. 1. Представление булевых операций для трёхмерных фигур

Особую актуальность приобретает задача выполнения логических операций в реальном времени, поскольку это открывает широкие возможности для динамической модификации объектов и интерактивного взаимодействия с виртуальной средой. Однако реализация таких операций предъявляет высокие требования к вычислительным ресурсам и производительности используемых алгоритмов. Эффективность выполнения логических операций напрямую зависит от выбранного способа представления трёхмерных объектов, алгоритмов обработки геометрии и особенностей программной платформы.

В настоящее время существуют различные подходы к описанию и представлению трёхмерных объектов, каждый из которых имеет свои преимущества

и ограничения при реализации булевых операций. Несмотря на наличие специализированных инструментов и библиотек, далеко не все существующие решения способны обеспечить требуемые производительность и качество выполнения операций в условиях реального времени.

Целью настоящего исследования были сравнительный анализ существующих методов и готовых реализаций для представления 3D-объектов и выполнения логических операций в реальном времени в среде игрового движка Unity и выработка рекомендаций по их эффективному использованию и оптимизации.

1. ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ

1.1. Полигональное представление объектов

Полигональная геометрия остаётся наиболее распространённым способом представления трёхмерных объектов, особенно в игровых движках. Объекты описываются с помощью вершин, рёбер и полигонов, формирующих сетку поверхности. Булевы операции в таких моделях выполняются за счёт геометрического пересечения полигонов, классификации граней и перестройки меша (см. рис. 2) на основе логических условий (объединение, пересечение, разность).

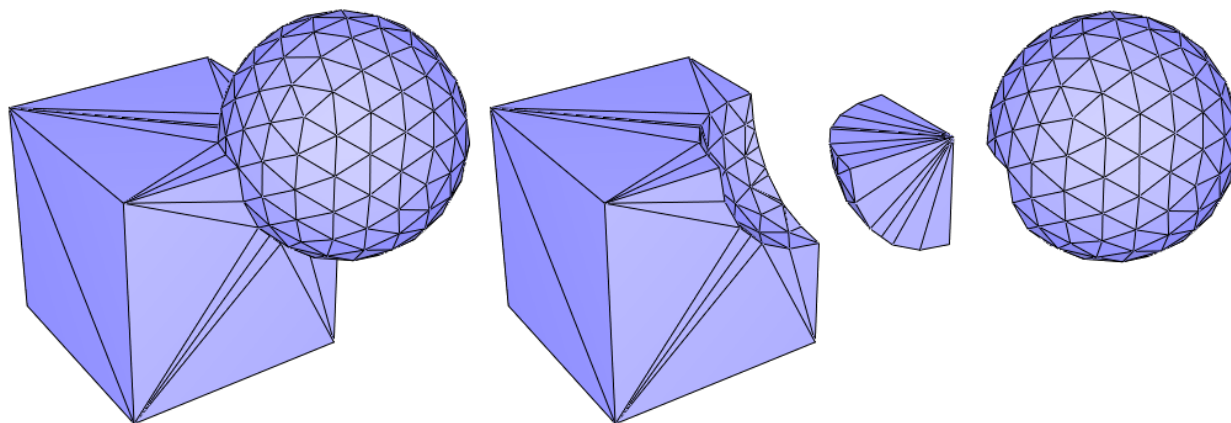


Рис. 2. Примеры операции вычитания над полигональными объектами

Преимущества полигонального представления объектов:

- возможность точной геометрии и контроля структуры модели;
- высокая совместимость с игровыми движками и стандартными средствами визуализации;
- поддержка анимации, освещения и текстурирования на уровне вершин и полигонов.

У полигонального представления имеются критичные недостатки:

- высокая чувствительность к качеству сетки – ошибки топологии, открытые грани;
- трудности с корректной обработкой самопересечений и пограничных случаев;
- высокая вычислительная сложность при перестроении полигональных сеток в реальном времени;
- необходимость в точном сопоставлении вершин, нормалей и UV-координат.

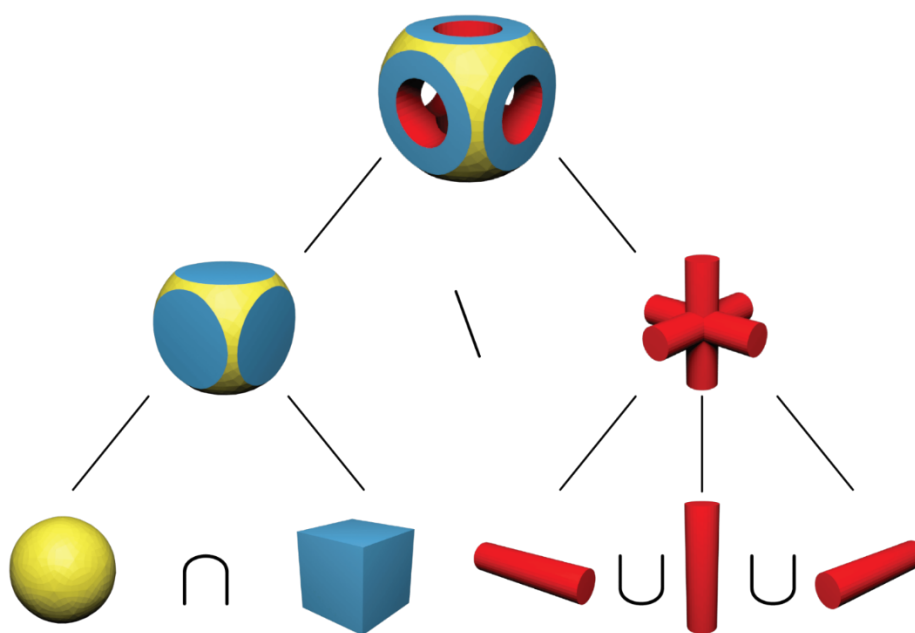


Рис. 3. Дерево булевых операций

Классическим примером реализации полигональной геометрии является конструктивная твердотельная геометрия (Constructive Solid Geometry, CSG), при которой булевы операции выражаются в виде дерева (см. рис. 3), где листьями выступают примитивы, а узлами – логические операции. Это метод Лейдлоу–Трампа–Хьюза, который был предложен в [2].

Итак, полигональные подходы, несмотря на широкую распространённость, предъявляют жёсткие требования к геометрии и требуют сложной обработки при выполнении булевых операций в реальном времени.

Математические основы конструктивной твердотельной геометрии и методы представления границ подробно рассмотрены в фундаментальной работе

[3]. Современные подходы к обеспечению робастности булевых операций [4] предлагают устойчивый алгоритм сегментации внутренних и внешних областей геометрических объектов.

1.2. Представление через поля расстояний

Функции знакового расстояния (Signed Distance Functions, SDF), другой часто используемый термин – «поля расстояний», представляют собой математический аппарат для описания трёхмерных объектов через скалярные поля, где каждая точка пространства характеризуется минимальным расстоянием до ближайшей поверхности объекта с учетом знака, указывающего положение точки относительно границы тела. Данный подход обеспечивает непрерывное и дифференцируемое представление геометрии, что критически важно для выполнения булевых операций в реальном времени без потери точности и с возможностью применения градиентных методов оптимизации. В отличие от дискретных представлений, таких как воксельные структуры или полигональные сетки, SDF сохраняют аналитическую точность на всех масштабах и позволяют эффективно реализовывать сложные топологические преобразования посредством алгебраических операций над скалярными функциями.

Концепция неявных поверхностей и методы их визуализации восходят к работе [5] по гипертекстурам. Метод трассировки сфер (Sphere tracing) как основной метод рендеринга SDF был формализован в [6]. Кроме того необходимо отметить всеобъемлющий обзор применений полей расстояний в компьютерной графике, который представлен в [7].

Этот подход предполагает описание формы объекта функцией расстояния до поверхности, что позволяет выполнять булевы операции в аналитическом виде: объединение реализуется как минимум двух значений, пересечение – максимум, а разность – как максимум первого и отрицания второго.

Отметим явные преимущества SDF-подхода:

- простота реализации булевых операций через математические выражения;
- хорошая масштабируемость при реализации на GPU;
- возможность использования сглаживающих функций для плавных переходов между объектами;

- поддержка процедурной геометрии и деформаций на лету.

Недостатки SDF-подхода:

- несовместимость со стандартными графическими конвейерами и необходимость применения объемных методов рендеринга (ray marching), исключающих использование аппаратно-ускоренной растеризации;
- ограниченная точность визуального отображения деталей при низком разрешении;
- сложность интеграции с физическим взаимодействием и коллизиями;
- требуется перепроектирование освещения и теней под объемное представление.

SDF-подход особенно привлекателен в задачах процедурной генерации и моделирования органических форм, но требует нетривиальной интеграции с классическим пайплайном визуализации. Дополнительным преимуществом является возможность применения сглаживающих функций (smooth-min) [8], создающих плавные переходы между формами (см. рис. 4). Это делает SDF-подход особенно привлекательным для визуально мягких и динамически изменяемых объектов [9].

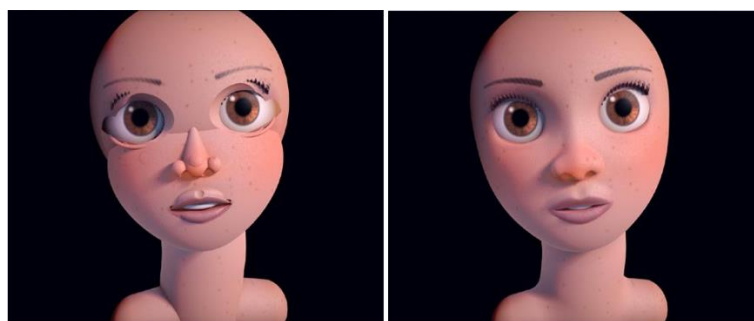


Рис. 4. Результаты проведения булевых функций [9]:

а) без сглаживающих функций; б) со сглаживающими функциями

Вместе с тем, как было отмечено ранее, визуализация таких моделей требует специальных алгоритмов, например ray marching [10], что ограничивает их применение в классическом рендеринге с полигональной сеткой.

1.3. Воксельные методы

Воксельные представления моделируют объекты как массив равномерных ячеек в трёхмерном пространстве. Булевы операции реализуются побитово, такой подход удобен для реализации разрушения, модификации или генерации сред в стиле игр Teardown или Minecraft.

Классическим алгоритмом визуализации воксельных данных является метод «марширующих кубов» (Marching Cubes) [11], обеспечивающий эффективное преобразование объемных данных в полигональные поверхности.

Преимущества воксельного представления:

- простота логических операций и устойчивость к топологическим ошибкам;
- естественная интеграция с физическим разрушением и симуляцией жидкостей;
- подходит для визуализации стилизованных миров (low-poly¹, cubic design²).

Недостатки этого подхода:

- высокая нагрузка на память при высокой детализации;
- необходимость последующей полигонализации для традиционного рендеринга;
- потери точности при преобразовании из других форматов;
- сложности с анимацией и точным управлением геометрией.

Воксельные подходы хорошо работают в сценариях, ограниченных по стилю или техническим условиям, но плохо масштабируются на реалистичную графику. Основными недостатками являются высокая нагрузка на память и ограничение по разрешению. Кроме того, детализация моделей ограничена размером вокселя, а реализация визуализации требует последующей полигонализации или специфического шейдерного рендеринга.

¹ Low-poly – стиль 3D-моделирования, характеризующийся использованием минимального количества полигонов для создания геометрических форм. Отличается упрощенными, угловатыми поверхностями и намеренно видимой полигональной структурой объектов.

² Cubic design — стилистический подход в 3D-графике, основанный на кубических и прямоугольных формах с четкими гранями и прямыми углами. Широко применяется в играх типа Minecraft и архитектурной визуализации для создания стилизованных блочных миров.

1.4. CAD-системы и инженерные решения

Булевы операции широко используются в инженерных CAD-системах (AutoCAD, SolidWorks, Fusion 360 и др.), где модели представлены в виде точных поверхностей и твёрдых тел, основанных на методе представления границ (boundary representation, B-Rep) и NURBS³. Эти системы обеспечивают высокую точность, контроль над параметризацией и удобство для последующего производства.

Однако CAD-подходы ориентированы преимущественно на офлайн-обработку и не адаптированы для выполнения логических операций в реальном времени. Кроме того, их форматы требуют конверсии при интеграции в игровые движки, что может приводить к потере точности или увеличению числа полигонов. Несмотря на это, принципы, используемые в CAD, легли в основу многих CSG-реализаций и алгоритмов булевых операций на уровне геометрических примитивов.

1.5. Сравнительный анализ

На основании проведенного анализа (табл. 1) можно заключить, что выбор подхода зависит от целей приложения.

³ NURBS (Non-Uniform Rational B-Splines) – математический метод представления кривых и поверхностей в компьютерной графике и CAD-системах, основанный на рациональных B-сплайнах с неравномерным распределением узловых точек. Обеспечивает точное описание как простых геометрических примитивов (окружности, эллипсы), так и сложных органических форм с высокой степенью гладкости и возможностью локального управления формой через контрольные точки.

Таблица 1. Сравнение рассмотренных подходов

Подход	Гибкость	Производительность	Визуальное качество	Сложность реализации	Работа в реальном времени
Полигональный	Средняя	Низкая	Высокое (при корректной сетке)	Высокая	Ограниченная
SDF	Высокая	Высокая (на GPU)	Высокое (smooth union)	Средняя	Хорошая
Воксели	Средняя	Средняя	Среднее	Средняя	Возможна
CAD-системы	Низкая	Низкая	Высокое (точные поверхности)	Высокая	Отсутствует

В условиях интерактивного применения и ограничений по производительности наибольший интерес представляют два направления: полигональные модели, как традиционный и точный способ описания геометрии, и функции знакового расстояния (SDF), как современный метод, обеспечивающий гибкость и высокую производительность при визуализации.

2. МЕТОДОЛОГИЯ ИССЛЕДОВАНИЯ

Для исследования был взят метод эксперимента: проследить за скоростью создания системы представления, реализованной самостоятельно, и сравнить результаты. В качестве кандидатов на тестирование были выбраны подходы полигональной сетки и SDF-функций.

Кроме того, помимо эмпирических исследований, был проведен анализ алгоритмов и систем для того, чтобы выявить нефункциональные свойства (не поддающиеся измерениям).

2.1. Используемые алгоритмы и теоретическое обоснование

Полигональный подход

Для реализации булевых операций над полигональными объектами была разработана модульная система-надстройка, обеспечивающая гибкое взаимодействие между компонентами геометрической модели и эффективную передачу данных между уровнями иерархии представления.

В качестве базового алгоритма выполнения булевых операций выбран метод Лейдлоу–Трампа–Хьюза [2], который обеспечивает точное пересечение полиэдральных объектов через классификацию результирующих граней на основе методов вычислительной геометрии. Выбор данного алгоритма обусловлен его открытой доступностью и возможностью последующей оптимизации для задач реального времени. Для повышения эффективности пространственных запросов были использованы принципы организации динамических иерархий ограничивающих объемов [12].

SDF-подход

Реализация функций знакового расстояния основана на библиотеке Unity Ray Marching [13], которая интегрирует современные достижения в области SDF-моделирования. Математические основы используемых функций расстояния базируются на работах Quilez [14] по трёхмерным SDF-примитивам и методам их композиции.

Ключевой особенностью реализации является поддержка сглаживающих операций, позволяющих создавать плавные переходы между геометрическими объектами при выполнении булевых операций. Для организации сложных композиций SDF-функций применяется древовидная структура на основе «ограничивающий параллелепипед, выровненный по осям» (axis-aligned bounding box, AABB), обеспечивающая эффективную пространственную локализацию вычислений [15].

Теоретическое обоснование SDF-подхода опирается на фундаментальные исследования [7] по применению полей расстояний в компьютерной графике, а также на современные методы процедурного моделирования с использованием неявных поверхностей.

2.2. Реализация подходов в среде Unity

Для реализации SDF-функций была использована открытая библиотека [13].

Для полигональных операций была разработана специализированная система представления геометрических данных, основанная на унифицированном интерфейсе доступа к топологическим связям.

Архитектура системы (см. рис. 5) построена на четырехуровневой иерархии компонентов:

Корневой узел (`ComplexMesh`) – содержит коллекцию полигонов и обеспечивает глобальный доступ к структуре модели;

- **Полигоны** – инкапсулируют три ребра и поддерживают обратную ссылку на корневой узел;
- **Рёбра** – связывают пары вершин и хранят ссылки на смежные полигоны;
- **Вершины** – представляют геометрические точки с информацией о принадлежности к рёбрам.

Центральным элементом архитектуры является универсальный метод `GetRelatedComponents<T>(bool isUpwards)`, обеспечивающий эффективную навигацию по иерархии представления. Параметр `isUpwards` определяет направление поиска: при значении `true` выполняется обращение к родительским компонентам (например, получение корневого узла от вершины), при `false` – к дочерним элементам (получение вершин полигона).

Построение топологической структуры осуществлялось методом `ConstructMeshFromBottom()` по принципу восходящей иерархии: последовательное создание вершин, формирование рёбер на основе треугольной информации `Unity Mesh`, объединение рёбер в полигоны и агрегация всех полигонов в единую коллекцию корневого узла.

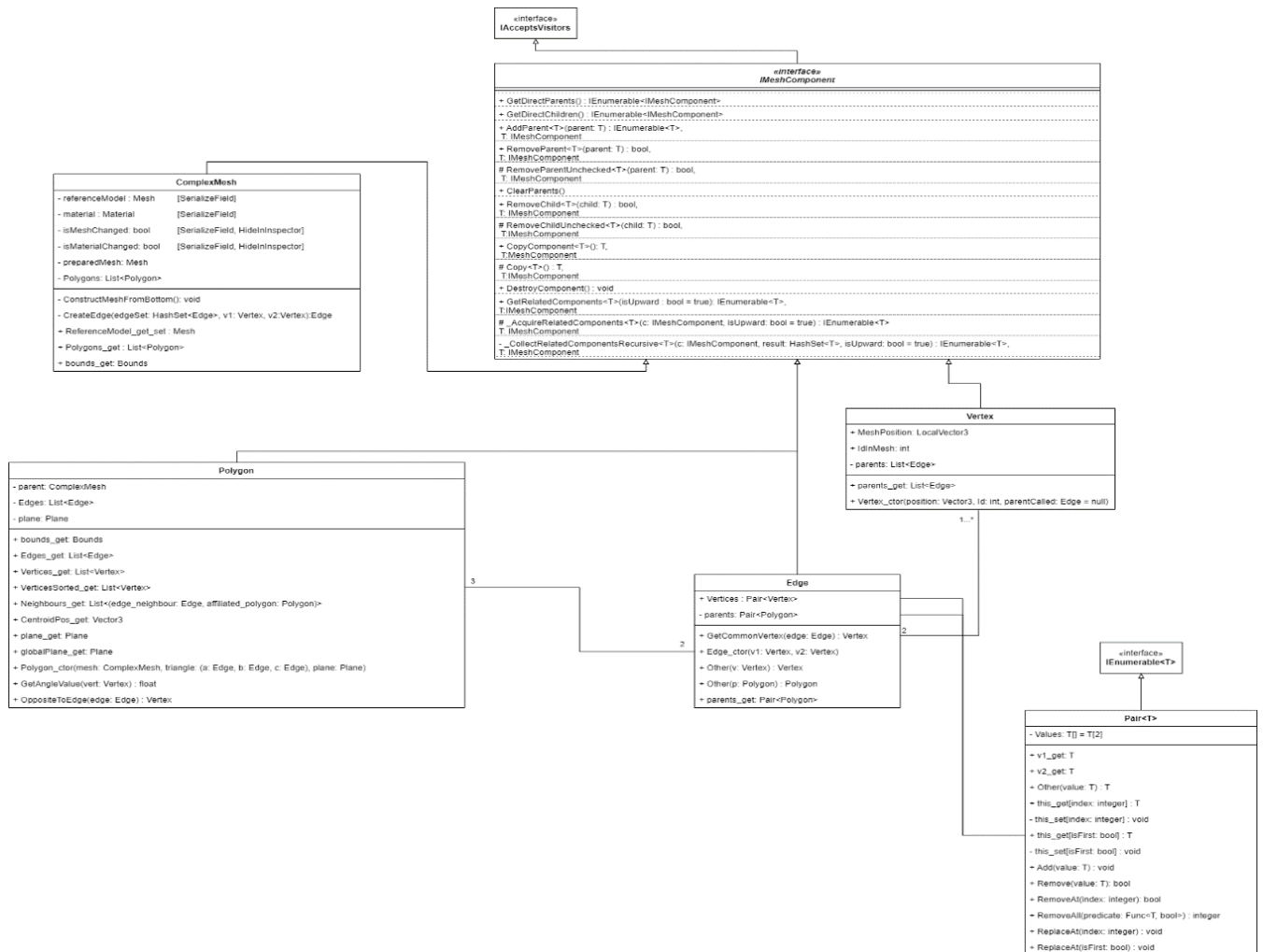


Рис. 5. Диаграмма классов разработанной системы

Каждый компонент содержит специализированную вычислительную информацию: полигоны хранят уравнения плоскостей, вершины – локальные координаты, что обеспечивает эффективность геометрических расчётов при выполнении булевых операций.

3. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

3.1. Описание тестовых моделей и среды

Для проведения сравнительного анализа был разработан единый программный модуль, включающий обе исследуемые библиотеки. В рамках эксперимента были созданы две идентичные по структуре сцены, в каждой из которых осуществлялось создание одного или нескольких объектов с использованием различных библиотек.

Измерение производительности выполнялось путём интеграции в исходный код функции таймера, осуществляющей точный замер временных интервалов, затрачиваемых на процесс построения трёхмерных моделей. Полученные временные показатели подвергались сравнительному анализу для определения эффективности каждой из исследуемых библиотек.

Выбор тестовых примитивов основан на стандартных практиках оценки производительности геометрических алгоритмов, где базовые формы (куб, сфера, цилиндр) обеспечивают воспроизводимость результатов и возможность сравнения с другими исследованиями [16].

3.2. Проведение экспериментов

В рамках исследования были выполнены тесты производительности для двух категорий геометрических объектов: полигональные сетки и SDF-функции.

Для анализа полигональных сеток использовались следующие тестовые модели:

1. Куб – базовый примитив, встроенный в среду разработки Unity;
2. Куб – модель, созданная в программе Autodesk Maya (для сравнения производительности при работе с встроенными примитивами и пользовательскими моделями, импортированными из внешних программных пакетов);
3. Цилиндр – стандартный геометрический примитив;
4. Сфера – сферический примитив;
5. Капсула – составной геометрический объект.

Поскольку для SDF-функций отсутствуют готовые примитивы, была проведена разработка соответствующих математических формул для создания аналогичных геометрических форм. Для каждого типа объекта были определены и реализованы математические выражения, позволяющие генерировать эквивалентные геометрические структуры посредством функций знакового расстояния [14].

Математические выражения для SDF-примитивов основаны на стандартных функциях расстояния, систематизированных в работах по неявному моделированию [6, 7].

3.3. Результаты экспериментов

Процедура измерения осуществлялась автоматически при назначении меша в соответствующее поле компонента. Встроенный в систему таймер фиксировал временные показатели выполнения операций. На основе полученных данных формировались графические представления результатов (рис. 5) для последующего анализа производительности обеих технологий.

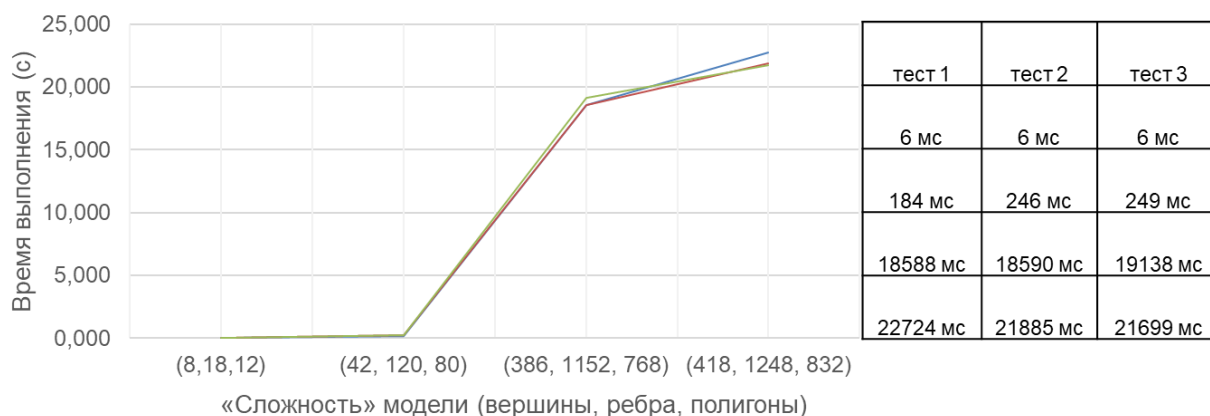


Рис. 5. Результаты построения системы полигональной сетки

Анализ производительности построения систем

Экспериментальные измерения показали существенные различия в характеристиках инициализации между исследуемыми подходами. Полигональные системы демонстрируют линейную зависимость времени построения от сложности геометрии, при этом время инициализации варьируется от 6 мс для базового куба до 21699 мс для сложных (см. рис. 5).

SDF-функции характеризуются константным временем инициализации, независимым от геометрической сложности описываемых объектов, поскольку представляют собой аналитические выражения без дискретных структур данных.

Анализ устойчивости результатов операций

Критическим различием между подходами является поведение системы при динамических изменениях геометрии. Полигональные представления обеспечивают сохранение результатов булевых операций: однажды выполненная операция сохраняет свой результат независимо от последующих трансформаций исходных объектов. Данное свойство обеспечивается фиксацией результирующей топологии в момент выполнения операции.

SDF-подход демонстрирует принципиально иное поведение: результат булевой операции вычисляется динамически для каждой пространственной точки на основе текущего положения операндов. Это приводит к тому, что перемещение одного из объектов после выполнения операции эффективно отменяет её результат, поскольку SDF-функция пересчитывает расстояния относительно новых позиций объектов. Выявленные ограничения SDF-подхода согласуются с теоретическими предсказаниями о природе функций знакового расстояния и их зависимости от пространственного контекста [7].

Влияние на долговременную производительность

Для решения проблемы непостоянства SDF-результатов требуется копирование состояния объектов на каждом кадре, что приводит к линейному росту вычислительных затрат со временем выполнения. Полигональные системы, напротив, демонстрируют стабильную производительность после завершения инициализации, поскольку результаты операций сохраняются в явном виде без необходимости пересчета.

Полигональные модели, несмотря на значительную вычислительную нагрузку при инициализации системы и выполнении алгоритма Лейдлоу–Трампа–Хьюза, обладают критически важным свойством устойчивого сохранения результатов операций. После завершения булевой операции система демонстрирует стабильную производительность без деградации со временем, поскольку результаты фиксируются в топологической структуре и не требуют пересчета при последующих трансформациях.

4. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Проведённые исследования выявили различные области применения рассматриваемых подходов в зависимости от временных требований задач. Полученные результаты согласуются с теоретическими предсказаниями о вычислительной сложности различных представлений трёхмерной геометрии [3, 6].

Традиционные CSG-алгоритмы с полигональными представлениями характеризуются высокими начальными затратами на построение и инициализацию системы. Однако данные методы поддаются существенной оптимизации и де-

монстрируют эффективность при выполнении длительных вычислительных операций, где время инициализации амортизируется за счёт продолжительности работы.

SDF-функции, напротив, обеспечивают высокую скорость выполнения операций изначально и не требуют сложной настройки системы. Тем не менее их применение ограничено краткосрочными задачами из-за накапливающихся вычислительных затрат при длительной работе.

Дополнительные возможности SDF-подхода проявляются в специализированных задачах многослойного рендеринга. В частности, при работе с объектами, состоящими из нескольких материальных слоев, SDF-функции позволяют эффективно определять принадлежность точек пересечения лучей к конкретным слоям геометрии. Это достигается путём вычисления знакового расстояния для каждой функции в точке пересечения и сопоставления результата с соответствующими текстурными данными, что особенно востребовано в задачах физического моделирования и визуализации сложных материалов.

5. ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ ДАЛЬНЕЙШИХ ИССЛЕДОВАНИЙ

Результаты проведенного исследования указывают на несколько перспективных направлений для повышения эффективности систем представления трёхмерных объектов при выполнении булевых операций в реальном времени.

5.1. Оптимизация структур данных для полигональных представлений

Текущая реализация полигональной системы демонстрирует значительные возможности для оптимизации в области управления топологическими связями. В настоящее время система использует динамические запросы для построения иерархических отношений между геометрическими примитивами (полигон → рёбра → вершины), что приводит к избыточным вычислениям при каждом обращении к структуре данных.

Классические подходы к оптимизации топологических структур в полигональном моделировании [3] могут быть адаптированы для современных требований интерактивных приложений. Основным направлением оптимизации является внедрение системы кэширования топологических связей с предварительным

вычислением отношений родительских и дочерних элементов. Такой подход позволит исключить повторные обходы коллекций и устранить дублирование общих вершин и рёбер при обработке смежных полигонов. Однако реализация данного механизма в условиях пок кадрового обновления требует разработки эффективных алгоритмов инвалидации кэша и управления жизненным циклом кэшированных данных.

5.2. Гибридные подходы к представлению геометрии

Перспективным направлением является исследование гибридных методов, где полигональные представления используются для основной геометрии, а SDF-функции применяются для определения многослойных материальных свойств объектов. Такая архитектура позволит сочетать точность полигональных моделей с гибкостью SDF-подхода при реализации сложных материальных структур и процедурных текстур.

5.3. Адаптивные алгоритмы уровня детализации

Дальнейшие исследования должны быть направлены на разработку адаптивных систем управления уровнем детализации (Level of Detail), способных автоматически регулировать сложность геометрических представлений в зависимости от вычислительных ресурсов и требований к качеству визуализации.

ЗАКЛЮЧЕНИЕ

Проанализированные системы представления трёхмерных объектов демонстрируют значительный потенциал для создания реалистичных симуляций динамического изменения геометрии в виртуальном пространстве.

Применение операций над полигональными мешами обеспечивает получение физически корректных и устойчивых геометрических преобразований с возможностью создания новых топологических структур. Данный подход эффективен для выполнения конструктивных операций, таких как булево вычитание, экструзия и деформация сложных объектов.

Использование SDF-функций открывает дополнительные возможности для реализации многослойного представления материалов в интерактивных системах. Например, при моделировании процесса среза древесного объекта система способна корректно отображать внутреннюю структуру материала, демонстрируя

переход от коры к древесине с соответствующими текстурными характеристиками.

Практическое применение разработанных методов охватывает широкий спектр специализированных симуляторов: от медицинских тренажеров хирургических операций (см., например, [17, 18]) до интерактивных «песочниц». Кроме того, предложенные решения могут быть интегрированы в современные игровые движки в качестве библиотек для динамического редактирования импортированной геометрии, что особенно востребовано в процессе разработки уровней, где требуется адаптация базовых моделей под специфические дизайнерские требования.

СПИСОК ЛИТЕРАТУРЫ

1. *Requicha A.G.* Representations for rigid solids: Theory, methods, and systems // ACM Computing Surveys (CSUR). 1980. Vol. 12, No. 4. P. 437–464.
2. *Laidlaw D.H., Trumbore W.B., Hughes J.F.* Constructive solid geometry for polyhedral objects // Proceedings of the 13th annual conference on Computer graphics and interactive techniques. 1986. P. 161–170.
3. *Mäntylä M.* An introduction to solid modeling. Computer Science Press, Inc., 1987.
4. *Jacobson A., Kavan L., Sorkine-Hornung O.* Robust inside-outside segmentation using generalized winding numbers // ACM Transactions on Graphics (TOG). 2013. Vol. 32, No. 4. P. 1–12.
5. *Perlin K., Hoffert E.M.* Hypertexture // Proceedings of the 16th annual conference on Computer graphics and interactive techniques. 1989. P. 253–262.
6. *Hart J. C.* Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces // The Visual Computer. 1996. Vol. 12, No. 10. P. 527–545.
7. *Jones M.W., Baerentzen J.A., Sramek M.* 3D distance fields: A survey of techniques and applications // IEEE Transactions on visualization and Computer Graphics. 2006. Vol. 12, No. 4. P. 581–599.
8. *Quilez I.* Smooth Minimum. 2013. URL: <https://iquilezles.org/articles/smin>
9. *Quilez I.* Smooth Minimum on ShaderToy // ShaderToy. URL: <https://www.shadertoy.com/view/DIVcW1>
10. *Walczyk M.* Ray Marching. Personal Blog.

URL: <https://michaelwalczyk.com/blog-ray-marching.html>

11. Lorensen W.E., Cline H.E. Marching cubes: A high resolution 3D surface construction algorithm // *Seminal graphics: pioneering efforts that shaped the field*. 1998. P. 347–353.

12. Zhou K. et al. Data-parallel octrees for surface reconstruction // *IEEE Transactions on visualization and computer graphics*. 2010. Vol. 17, No. 5. P. 669–681.

13. Chou M.-L. Unity Ray Marching // GitHub.

URL: <https://github.com/TheAltenChou/unity-ray-marching>

14. Quilez I. 3D SDFs. 2013. URL: <https://iquilezles.org/articles/distfunctions/>

15. Catto E. Dynamic Bounding Volume Hierarchies // *Game Developers Conference*. 2019. URL: https://box2d.org/files/ErinCatto_DynamicBVH_GDC2019.pdf

16. Bernstein G., Fussell D. Fast, exact, linear booleans // *Computer Graphics Forum*. Oxford, UK: Blackwell Publishing Ltd, 2009. Vol. 28, No. 5. P. 1269–1278.

17. Шареева Р.А. и др. Подходы к проектированию виртуальных тренажёров хирургических операций // *Электронные библиотеки*. 2022. Т. 25, №. 5. С. 489–532.

18. Кузупакова В.В. и др. Virtual surgery system with realistic visual effects and haptic interaction // *Proc. of The International Conference on Artificial Life And Robotics*. 2017. С. P86–P89.

3D OBJECTS REPRESENTATION FOR REAL-TIME BOOLEAN OPERATIONS

I. E. Plotnikov¹ [0009-0004-2891-9913], **D. I. Kostyuk**² [0000-0002-6542-6980]

^{1, 2}*Kazan Federal University, Kazan, 420008, Russia*

¹ilaya.link@gmail.com, ²xdxnkx@gmail.com

Abstract

The paper presents a comparative analysis of methods for representing three-dimensional objects to perform real-time Boolean operations in the Unity game engine environment. Four main approaches are considered: polygonal representation based on constructive solid geometry (CSG), sign distance functions (SDF), voxel methods and CAD-systems with boundary representation (B-Rep) and NURBS-surfaces.

An experimental study of the performance of polygonal algorithms of Boolean operations and SDF functions based on ray marching implementation is carried out. It is revealed that polygonal methods are characterized by high initial system construction costs, but provide stable performance during long operations and preservation of transformation results. SDF functions demonstrate high speed of operations and flexibility in creating smooth transitions between objects, but are limited in application for long-term tasks due to the peculiarities of the computational model.

The areas of effective application of each approach are identified: polygonal methods are recommended for tasks requiring precise geometric control and integration with traditional graphics pipelines, while SDF functions are optimal for procedural generation, multilayer material rendering and creation of dynamic visual effects. The results of the study can be used in the development of interactive simulators, game applications and virtual reality systems.

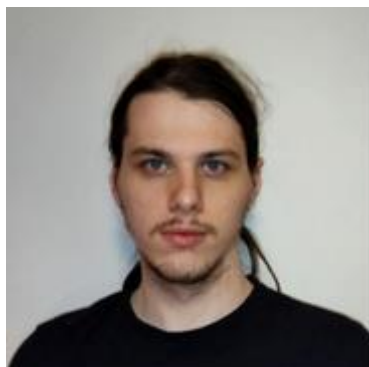
Keywords: *boolean operations, 3D modeling, constructive solid geometry, sign distance functions, SDF, Unity, real-time, polygonal meshes, ray marching.*

REFERENCES

1. Requicha A.G. Representations for rigid solids: Theory, methods, and systems // ACM Computing Surveys (CSUR). 1980. Vol. 12, No. 4. P. 437–464.
2. Laidlaw D.H., Trumbore W.B., Hughes J.F. Constructive solid geometry for polyhedral objects // Proceedings of the 13th annual conference on Computer graphics and interactive techniques. 1986. P. 161–170.
3. Mäntylä M. An introduction to solid modeling. Computer Science Press, Inc., 1987.
4. Jacobson A., Kavan L., Sorkine-Hornung O. Robust inside-outside segmentation using generalized winding numbers // ACM Transactions on Graphics (TOG). 2013. Vol. 32, No. 4. P. 1–12.
5. Perlin K., Hoffert E.M. Hypertexture // Proceedings of the 16th annual conference on Computer graphics and interactive techniques. 1989. P. 253–262.
6. Hart J. C. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces // The Visual Computer. 1996. Vol. 12, No. 10. P. 527–545.

7. Jones M.W., Baerentzen J.A., Sramek M. 3D distance fields: A survey of techniques and applications // IEEE Transactions on visualization and Computer Graphics. 2006. Vol. 12, No. 4. P. 581–599.
8. Quilez I. Smooth Minimum. 2013. URL: <https://iquilezles.org/articles/smin>
9. Quilez I. Smooth Minimum on ShaderToy // ShaderToy.
URL: <https://www.shadertoy.com/view/DIVcW1>
10. Walczyk M. Ray Marching. Personal Blog.
URL: <https://michaelwalczyk.com/blog-ray-marching.html>
11. Lorensen W.E., Cline H.E. Marching cubes: A high resolution 3D surface construction algorithm // Seminal graphics: pioneering efforts that shaped the field. 1998. P. 347–353.
12. Zhou K. et al. Data-parallel octrees for surface reconstruction // IEEE Transactions on visualization and computer graphics. 2010. Vol. 17, No. 5. P. 669–681.
13. Chou M.-L. Unity Ray Marching // GitHub.
URL: <https://github.com/TheAllenChou/unity-ray-marching>
14. Quilez I. 3D SDFs. 2013. URL: <https://iquilezles.org/articles/distfunctions/>
15. Catto E. Dynamic Bounding Volume Hierarchies // Game Developers Conference. 2019. URL: https://box2d.org/files/ErinCatto_DynamicBVH_GDC2019.pdf
16. Bernstein G., Fussell D. Fast, exact, linear booleans // Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2009. Vol. 28, No. 5. P. 1269–1278.
17. Sharaeva R. A. et al. Approaches to the design of virtual simulators of surgical operations // Russian Digital Libraries. 2022. Vol. 25, No. 5. P. 489–532.
18. Kugurakova V. et al. Virtual surgery system with realistic visual effects and haptic interaction // Proc. of The International Conference On Artificial Life And Robotics. 2017. P. P86–P89.

СВЕДЕНИЯ ОБ АВТОРАХ



ПЛОТНИКОВ Илья Евгеньевич – студент 2 курса магистратуры Института информационных технологий и интеллектуальных систем Казанского федерального университета. Сфера научных интересов: логические операции над твердотельными трехмерными объектами в реальном времени.

Ilya Evgenievich PLOTNIKOV – 2nd year master's student at the Institute of Information Technologies and Intelligent Systems of Kazan Federal University. Research interests: logical operations on solid three-dimensional objects in real time.

email: ilaya.link@gmail.com

ORCID: 0009-0004-2891-9913



КОСТЮК Даниил Иванович – старший преподаватель кафедры индустрии разработки видеоигр. Сфера научных интересов: процедурная генерация сцен, моделирование, рендеринг.

Daniil Ivanovich KOSTYUK – senior lecturer of the Department of Video Game Development Industry. Research interests include procedural scenes generation, modeling, rendering.

email: xdxnxkx@gmail.com

ORCID: 0000-0002-6542-6980

Материал поступил в редакцию 2 апреля 2025 года