

В современных видеоиграх и приложениях реального времени сцены могут содержать тысячи или даже миллионы объектов. Без отсека система будет пытаться рендерить все объекты, даже если большая часть из них не видна. Это значительно замедлит процесс рендеринга и снизит производительность.

Culling (с англ. отсев, отбор, выборка) — процесс оптимизации рендеринга, суть которого заключается в исключении невидимых объектов в кадре, с целью экономии вычислительных ресурсов.

Отсев (Culling) позволяет уменьшить нагрузку на центральный и графический процессоры, избегая рендеринга ненужных объектов. Это особенно критично для игр с большими открытыми мирами, таких как RPG или FPS, где игрок перемещается по огромным пространствам с множеством объектов.

└ Culling

В современных видеоиграх и приложениях реального времени сцены могут содержать тысячи или даже миллионы объектов. Без отсека система будет пытаться рендерить все объекты, даже если большая часть из них не видна. Это значительно замедлит процесс рендеринга и снизит производительность.

Culling (с англ. отсев, отбор, выборка) — процесс оптимизации рендеринга, суть которого заключается в исключении невидимых объектов в кадре, с целью экономии вычислительных ресурсов.

Отсев (Culling) позволяет уменьшить нагрузку на центральный и графический процессоры, избегая рендеринга ненужных объектов. Это особенно критично для игр с большими открытыми мирами, таких как RPG или FPS, где игрок перемещается по огромным пространствам с множеством объектов.



Рис. 1: Открытый мир в игре Far Cry (2004 г.)

Этапы с применением отсева

- **Geometric stage (Геометрический этап).** Отсев начинается с анализа сцены, где объекты проверяются на видимость относительно камеры.
Frustum Culling и Portal Culling применяются на этом этапе.
- **Rasterization stage (Этап растеризации).** Позволяет исключить полигоны (поверхности), нормаль которых направлена от камеры (или к камере).
Этот этап включает Face Culling.
- **Pixel stage (Этап обработки пикселей).** Применяется, когда пиксели уже предварительно обработаны, если объекты перекрываются другими. Аппаратные и программные методы позволяют сделать это на уровне GPU, что помогает избежать затрат на рендеринг окклюдированных объектов.
На этом этапе применяется Occlusion Culling.

└ Этапы с применением отсева

Комплексные техники и оптимизации (требующие дополнительных вычислений)

Hierarchical Culling (Иерархический отсев) и Level of Detail (LOD)

Geometric stage. Frustum Culling

Отсев по пирамиде видимости

Простая идея. Если камера смотрит прямо, все объекты за спиной камеры будут исключены из обработки.

Пирамида видимости (frustum) — объём пространства, который видит камера в 3D-пространстве. Всё, что находится за пределами этой пирамиды, не будет видно в кадре.

Frustum Culling — процесс, при котором объекты, находящиеся за пределами пирамиды видимости камеры, исключаются из рендеринга. Это один из самых простых и эффективных методов отсечения.

2025-10-18

Отсев

Geometric stage. Frustum Culling

Geometric stage. Frustum Culling
Отсев по пирамиде видимости

Простая идея. Если камера смотрит прямо, все объекты за спиной камеры будут исключены из обработки.
Пирамида видимости (frustum) — объём пространства, который видит камера в 3D-пространстве. Всё, что находится за пределами этой пирамиды, не будет видно в кадре.
Frustum Culling — процесс, при котором объекты, находящиеся за пределами пирамиды видимости камеры, исключаются из рендеринга. Это один из самых простых и эффективных методов отсечения.

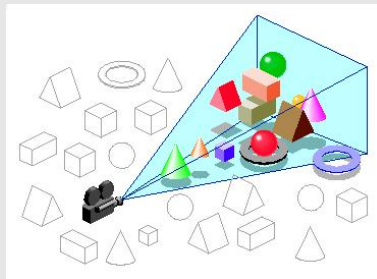
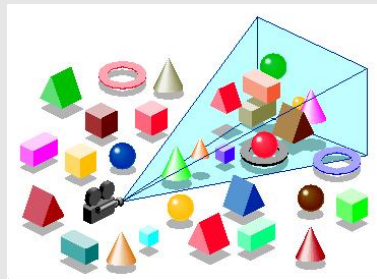


Рис. 2: Отсев по пирамиде видимости: до (слева) и после (справа)

Geometric stage. Portal Culling

Отсев через порталы

Portal Culling — метод, применяемый в сценах, которые делятся на секции или комнаты, соединенные «порталами» (двери, окна и т.д.). Если камера не может видеть через портал, то объекты, находящиеся в другой комнате, будут исключены. Этот метод часто используется в играх, где игрок перемещается между различными областями, например, в зданиях или туннелях.

Пример.

Пусть дана сцена квартиры, с заполненными комнатами.

Задача. Исключить объекты, находящиеся вне зоны видимости, т.е. за порталами (двери, окна).

Например, если дверь в другую комнату закрыта, сцена внутри другой комнаты может быть исключена из рендеринга.

2025-10-18

Отсев

Geometric stage. Portal Culling

Geometric stage. Portal Culling
Отсев через порталы

Portal Culling — метод, применяемый в сценах, которые делятся на секции или комнаты, соединенные «порталами» (двери, окна и т.д.). Если камера не может видеть через портал, то объекты, находящиеся в другой комнате, будут исключены. Этот метод часто используется в играх, где игрок перемещается между различными областями, например, в зданиях или туннелях.

Пример.
Пусть дана сцена квартиры, с заполненными комнатами.
Задача. Исключить объекты, находящиеся вне зоны видимости, т.е. за порталами (двери, окна).
Например, если дверь в другую комнату закрыта, сцена внутри другой комнаты может быть исключена из рендеринга.

Основное предназначение этого метода заключается в оптимизации отрисовки игр с архитектурными сценами.

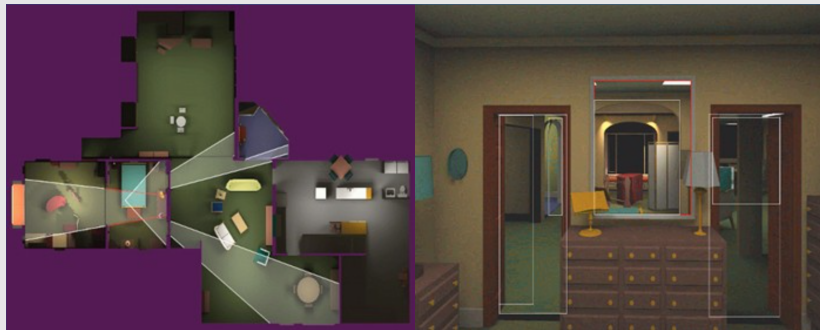


Рис. 3: Portals and Mirrors, David P. Luebke and Chris Georges

Rasterization stage. Face Culling

Исключение граней

Face culling (исключение граней) — более общий термин, который описывает процесс исключения каких-либо (например, обратных или лицевых) граней (полигонов) объекта из рендеринга.

Backface culling (исключение обратных граней) — техника, которая исключает из рендеринга задние (обратные) грани трёхмерных объектов. Эти грани считаются невидимыми для зрителя, так как они направлены в противоположную от камеры сторону.

Описание Backface culling:

- Трёхмерные объекты обычно состоят из полигонов (чаще всего треугольников), которые часто рассматриваются как однонаправленные поверхности.
- Каждый треугольник имеет нормаль (вектор, перпендикулярный поверхности).
- Если нормаль треугольника направлена от камеры (т.е. обратная сторона полигона), этот треугольник считается невидимым и исключается из рендеринга.

2025-10-18

Отсев

└ Rasterization stage. Face Culling

Rasterization stage. Face Culling

Исключение граней

Face culling (исключение граней) — более общий термин, который описывает процесс исключения каких-либо (например, обратных или лицевых) граней (полигонов) объекта из рендеринга.
Backface culling (исключение обратных граней) — техника, которая исключает из рендеринга задние (обратные) грани трёхмерных объектов. Эти грани считаются невидимыми для зрителя, так как они направлены в противоположную от камеры сторону.

Описание Backface culling:

- Трёхмерные объекты обычно состоят из полигонов (чаще всего треугольников), которые часто рассматриваются как однонаправленные поверхности.
- Каждый треугольник имеет нормаль (вектор, перпендикулярный поверхности).
- Если нормаль треугольника направлена от камеры (т.е. обратная сторона полигона), этот треугольник считается невидимым и исключается из рендеринга.

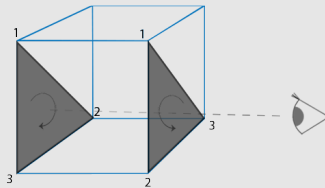


Рис. 4: Face culling: для куба, состоящего из треугольников (слева) и применение техники в игровой сцене The Sims (справа)

Pixel stage. Occlusion Culling

Отсев по окклюзии

Этот метод отсечения удаляет из рендеринга объекты, которые перекрыты другими объектами и полностью скрыты от камеры. Например, если за зданием находится дерево, и здание полностью его закрывает, дерево можно исключить из рендеринга. Occlusion Culling работает с учетом геометрии сцены, рассчитывая, какие объекты скрыты за другими.

Пример.
Стоя перед зданием, камера не увидит объекты, находящиеся за ним. Тогда можно не тратить вычислительные ресурсы на отрисовку этих объектов, так как они не повлияют на кадр.

2025-10-18
Отсев

Pixel stage. Occlusion Culling

Pixel stage. Occlusion Culling
Отсев по окклюзии

Этот метод отсечения удаляет из рендеринга объекты, которые перекрыты другими объектами и полностью скрыты от камеры. Например, если за зданием находится дерево, и здание полностью его закрывает, дерево можно исключить из рендеринга. Occlusion Culling работает с учетом геометрии сцены, рассчитывая, какие объекты скрыты за другими.

Пример.
Стоя перед зданием, камера не увидит объекты, находящиеся за ним. Тогда можно не тратить вычислительные ресурсы на отрисовку этих объектов, так как они не повлияют на кадр.

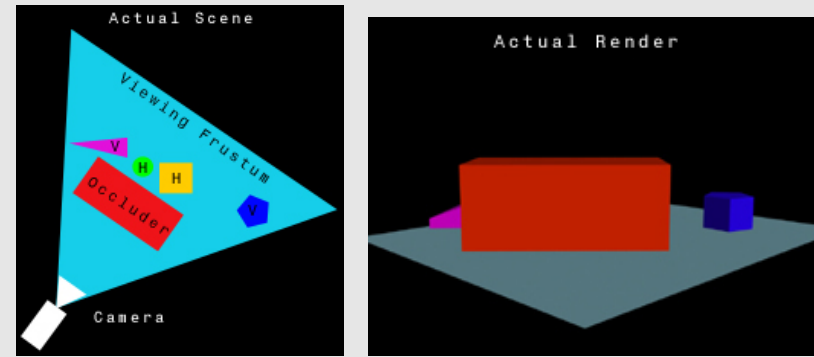


Рис. 5: Occlusion culling: пример сцены (слева) и результат работы (справа)

Hierarchical Culling

Иерархический отсев

Hierarchical Culling — оптимизация рендеринга путём исключения невидимых объектов на основе их положения и иерархической структуры.

Основные этапы:

- Построение иерархической структуры объектов сцены.
 - Bounding Volume. Каждая группа объектов оборачивается в объёмный контейнер (сферу или прямоугольный параллелепипед, AABB), который проверяется на пересечение с camera frustum камеры.
 - Иерархическая структура. Объекты группируются в древовидную структуру (например, Bounding Volume Hierarchy, BVH).
- Рекурсивный отсев при отрисовке сцены. Если объём не пересекается с frustum, вся группа объектов исключается. Если пересекается, проверяются дочерние объекты.

Примечание. AABB (Axis-Aligned Bounding Box) — прямоугольный ограничивающий объём, выровненный по осям координат.

2025-10-18

Отсев

└ Hierarchical Culling

Hierarchical Culling
Иерархический отсев

Hierarchical Culling — оптимизация рендеринга путём исключения невидимых объектов на основе их положения и иерархической структуры.

Основные этапы:

- Построение иерархической структуры объектов сцены.
 - Bounding Volume. Каждая группа объектов оборачивается в объёмный контейнер (сферу или прямоугольный параллелепипед, AABB), который проверяется на пересечение с camera frustum камеры.
 - Иерархическая структура. Объекты группируются в древовидную структуру (например, Bounding Volume Hierarchy, BVH).
- Рекурсивный отсев при отрисовке сцены. Если объём не пересекается с frustum, вся группа объектов исключается. Если пересекается, проверяются дочерние объекты.

Примечание. AABB (Axis-Aligned Bounding Box) — прямоугольный ограничивающий объём, выровненный по осям координат.

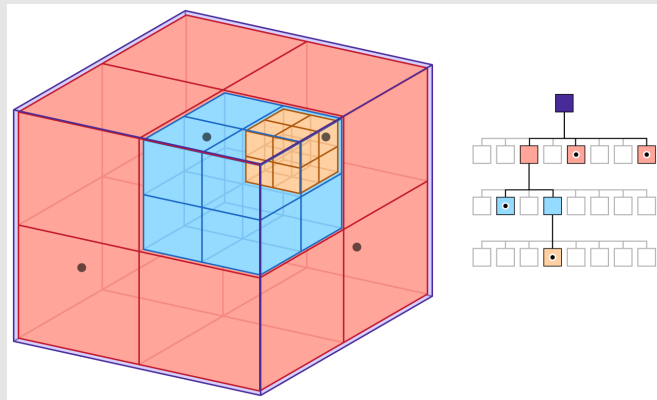


Рис. 6: Визуализация схемы пространственного разделения и структура октодеревя (Octree)

Level of Detail (LOD)

Уровень детализации

Level of Detail (уровень детализации) не является прямым видом отсечения, хотя часто используется вместе с отсечком для оптимизации рендеринга.

LOD (Level of Detail) — техника, при которой объекты рендерятся с меньшей детализацией, если они находятся далеко от камеры. Таким образом, уменьшается количество полигонов, обрабатываемых для удалённых объектов, сохраняя ресурсы.

Примечание.
Level of Development (уровень разработки) — термин, используемый в архитектуре и строительстве, особенно в контексте информационного моделирования зданий (Building Information Modeling, BIM). Он описывает степень проработки и детализации объектов модели в различных стадиях проектирования и строительства.

Mipmap (Multum In Parvo, Многое в малом) представляет собой предварительно созданный набор (иерархию) изображений, где каждое последующее изображение в этом наборе имеет размер в половину (или в другой пропорции) меньший, чем предыдущее.

2025-10-18

Отсев

Level of Detail (LOD)

Level of Detail (LOD)
Уровень детализации

Level of Detail (уровень детализации) не является прямым видом отсечения, хотя часто используется вместе с отсечком для оптимизации рендеринга.

LOD (Level of Detail) — техника, при которой объекты рендерятся с меньшей детализацией, если они находятся далеко от камеры. Таким образом, уменьшается количество полигонов, обрабатываемых для удалённых объектов, сохраняя ресурсы.

Примечание.
Level of Development (уровень разработки) — термин, используемый в архитектуре и строительстве, особенно в контексте информационного моделирования зданий (Building Information Modeling, BIM). Он описывает степень проработки и детализации объектов модели в различных стадиях проектирования и строительства.

Mipmap (Multum In Parvo, Многое в малом) представляет собой предварительно созданный набор (иерархию) изображений, где каждое последующее изображение в этом наборе имеет размер в половину (или в другой пропорции) меньший, чем предыдущее.

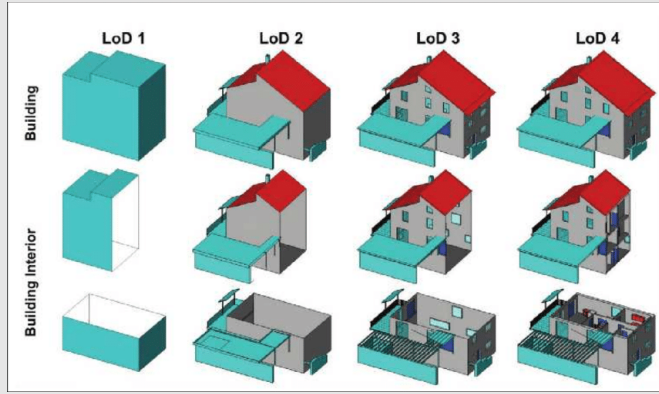


Рис. 7: Different levels of detail (LoD) for buildings. Source: Karlsruhe Institute of Technology (KIT), CityGML 2.0 Encoding standard

- Экономия вычислительных ресурсов (процессор, GPU)
- Уменьшение количества полигонов для рендеринга
- Улучшение производительности в реальном времени (FPS)

Сегодня Umbra используется в современных играх, таких как The Witcher 3, Call of Duty, и Assassin's Creed и другие.