

Методы исследования видимых поверхностей

09

КЛАССИФИКАЦИЯ АЛГОРИТМОВ ИССЛЕДОВАНИЯ ВИДИМЫХ ПОВЕРХНОСТЕЙ

- Алгоритмы исследования видимых поверхностей классифицируют согласно тому, что в них фигурирует определения объектов или их спроектированные образы.
- Подходы соответственно
 - методы пространства объектов
 - и методы пространства изображений.
- В методах первого класса объекты и части объектов сравниваются друг с другом в контексте определения сцены, и в результате некоторые поверхности помечаются как видимые.
- В алгоритмах отображения линий, например, для идентификации видимых линий каркасных изображений обычно используются методы пространства объектов.
- В алгоритмах пространства изображений видимость определяется для каждого положения пикселя на плоскости проекции.
- В большинстве алгоритмов выявления видимых поверхностей используются методы пространства изображений.
- Базовые подходы, принятые в различных алгоритмах выявления видимых поверхностей, могут довольно значительно отличаться, однако в большинстве для улучшения производительности используются методы сортировки и когерентности.
 - **Сортировка** облегчает сравнение по глубине, упорядочивая отдельные поверхности на сцене согласно их расстоянию от плоскости наблюдения.
 - Методы **когерентности** основаны на регулярности структур на сцене. Отдельная строка развертки очень часто содержит интервалы (серии) пикселей постоянной интенсивности, структура строки развертки зачастую сохраняется при переходе к следующей строке

ДЕТЕКТИРОВАНИЕ НЕВИДИМЫХ ПОВЕРХНОСТЕЙ

- Быстрый и простой **метод пространства объектов** для определения невидимых граней многогранника основан на проверках невидимых поверхностей.
- ✓ Точка (x, y, z) расположена за многоугольной поверхностью, если $Ax + By + Cz + D < 0$, где A, B, C и D параметры плоскости многоугольника.
- Для обнаружения задних граней можно использовать точку наблюдения.
- Для упрощения рассмотрим направление вектора нормали N к многоугольной поверхности. Если V_{view} вектор направления наблюдения от положения камеры, (рис. 1), то многоугольник – задняя грань, если $V_{view} \cdot N > 0$.
- Если описания объекта преобразованы в координаты проекции, а направление наблюдения параллельно оси z_v , требуется рассмотреть только z вектора нормали N .
- В правосторонней системе координат, (направление наблюдения - отрицательное направление оси z_v (рис.2)), многоугольник невидим, если компонент z (C) его вектора нормали N удовлетворяет соотношению $C < 0$. а также, грань, N которой имеет компонент z $C = 0$, не видна, т.к. направление наблюдения - касательная к многоугольнику.
- ✓ В общем случае любой многоугольник можно пометить как невидимую грань, если компонент z его вектора нормали удовлетворяет неравенству $C \leq 0$.

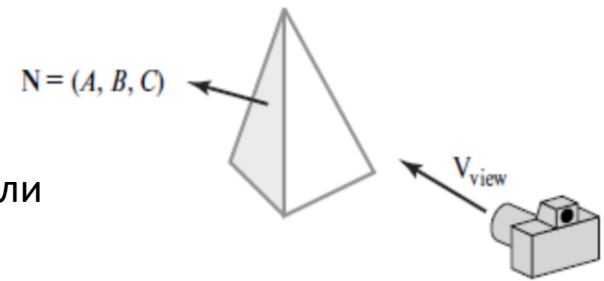


Рис. 1. Вектор нормали к поверхности N и вектор направления наблюдения V_{view}

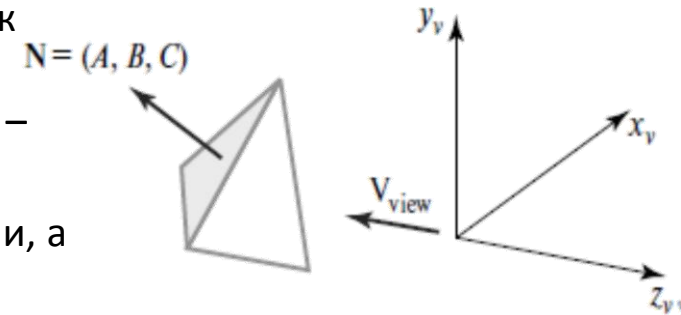


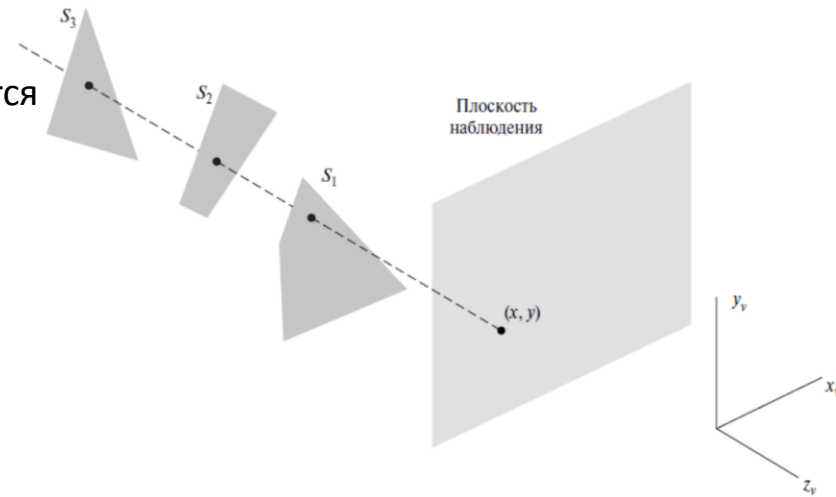
Рис.2. Многоугольная поверхность с параметром плоскости $C < 0$ в правосторонней системе координат идентифицируется как задняя грань, если направление наблюдения совпадает с отрицательным направлением оси z_v

ДЕТЕКТИРОВАНИЕ НЕВИДИМЫХ ПОВЕРХНОСТЕЙ

- Подобные методы можно использовать и если наблюдение ведется в левосторонней системе координат. Параметры плоскости A , B , C и D вычисляются по координатам вершин многоугольника, заданным в направлении по часовой стрелке (вместо против часовой, в правосторонней системе).
- Проверка $(Ax + By + Cz + D < 0)$ корректна и для точек, расположенных за многоугольником. Векторы нормалей невидимых граней направлены от точки наблюдения. Когда направление наблюдения совпадает с положительным направлением оси z_v , они определяются условием $C \geq 0$.
- Исследуя параметр C для различных плоских поверхностей, описывающих объект, можно непосредственно выявить все невидимые грани.
 - Для выпуклого многогранника, проверка определяет все скрытые поверхности на сцене, любая поверхность либо полностью видима, либо полностью невидима.
 - Если сцена содержит только неперекрывающиеся выпуклые многоугольники, то с помощью метода определения невидимых граней выявляются все скрытые поверхности.
 - Для объектов, таких как вогнутый многоугольник, необходимо больше проверок, чтобы определить наличие граней, полностью или частично загороженных другими гранями.
- Хороший алгоритм удаления невидимых граней должен отсеивать около половины многоугольных поверхностей на сцене.

МЕТОД БУФЕРА ГЛУБИНЫ

- 1 из подходов к выявлению видимых поверхностей в пространстве изображений - **метод буфера глубины**.
- Для каждого пикселя на плоскости проекции сравниваются значения глубин поверхностей на сцене. Каждая поверхность сцены обрабатывается отдельно, рассматриваются все положения пикселей.
- Метод **z-буфера** – глубина объекта измеряется по z .
- На рис. поверхности с разным расстоянием по линии ортогографической проекции от точки (x, y) . Их можно исследовать в любом порядке.
- Глубина поверхности от плоскости наблюдения сравнивается с глубинами ранее обработанных.
- Если поверхность ближе к точке наблюдения, ее цвет рассчитывается и записывается вместе с глубиной.
- Набор записанных цветов поверхности определяет видимые поверхности на сцене.
- Алгоритм реализуется в нормированных координатах, значения глубины - от 0 у ближней плоскости отсечения (плоскость наблюдения) до 1 у дальней.
- Требуется **две области буфера**. Глубина для каждой точки (x, y) записывается в буфере глубины, код цвета каждого пикселя – в буфер кадров.
- Изначально все позиции в буфере глубины = 1 (максимальная глубина), а в буфере кадров (регенерации) - цвет фона.



- Рис. 9.4. В один пиксель плоскости наблюдения (x, y) проектируются три точки наклаывающихся поверхностей. Наименьшее значение глубины имеет видимая поверхность S_1
- Затем обрабатываются все поверхности из таблиц многоугольников по 1-ой строке развертки за раз,
- для каждого пикселя (x, y) рассчитывается глубина и сравнивается со значением в буфере.
- Если < значения в буфере глубины, записывается новое значение.
- Цвет поверхности в точке вычисляется и помещается в позицию буфера кадра, соответствующую пикселю.

МЕТОД БУФЕРА ГЛУБИНЫ

- Значения глубины нормированы в промежуток 0-1,
- плоскости наблюдения соответствует глубина 0.
- Алгоритм можно применить для любого диапазона глубин.
- Алгоритм буфера глубины
 - 1. Инициализировать буфер глубины и буфер кадров так,
 - $\text{depthBuff}(x, y) = 1.0$,
 - $\text{frameBuff}(x, y) = \text{backgndColor}$.
 - 2. Обработать по одному все многоугольники сцены.
 - Для каждого положения пикселя (x, y) многоугольника вычислить
 - глубину z (если неизвестна).
 - Если $z < \text{depthBuff}(x, y)$, вычислить цвет поверхности в этой точке и положить
 - $\text{depthBuff}(x, y) = z$,
 - $\text{frameBuff}(x, y) = \text{surfColor}(x, y)$.
 - После обработки в буфере глубины значения глубин для видимых поверхностей, в буфере кадров соответствующие коды цвета для поверхностей.
- По глубинам вершин многоугольника на сцене можно вычислить глубину любой точки плоскости из уравнения плоскости. $z = (-Ax - By - D) / C$
- Для любой строки развертки соседнее горизонтальное положение x вдоль строки отличается от текущего на ± 1 , y соседних строк развертки на ± 1 . Если глубина точки (x, y) равна z , то глубина z' следующей точки $(x+1, y)$ вдоль строки развертки
- $z' = (-A(x+1) - By - D) / C$
- Или $z' = z - A / C$
- Отношение $A = C$ постоянно для каждой поверхности, так что последовательные значения глубин вдоль строки развертки находятся из предыдущих с помощью операции сложения.

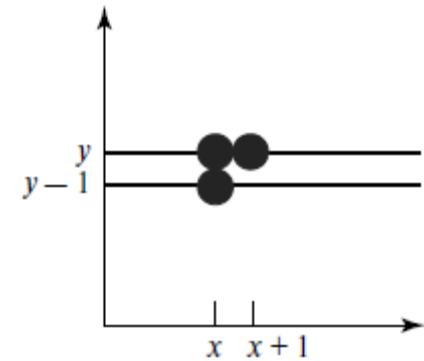


Рис. Для точки (x, y) на строке развертки следующая точка вдоль строки имеет координаты $(x + 1, y)$, а точка, расположенная непосредственно под данной, координаты $(x, y - 1)$

МЕТОД БУФЕРА ГЛУБИНЫ

- Обработка пикселей слева направо вдоль каждой строки развертки **начинается** с расчета глубины **левого** края многоугольника, пересекающего эту строку развертки (рис. 1).
- Затем для каждого последующего положения вдоль строки развертки рассчитывается значение глубины.
- Начать можно с верхней вершины, затем значения координат x рекурсивно вычисляются вниз по левой стороне многоугольника – x начальной точки в каждой строке вычисляется из начального (краевого) значения x в предыдущей строке развертки.
- $x' = x - 1/m$,
- где m наклон этой стороны (рис. 2).
- $z' = z + (A/m + B)/C$
- Для вертикальной стороны крутизна (наклон)
- равна бесконечности, т.е.
- $z' = z + B/C$.

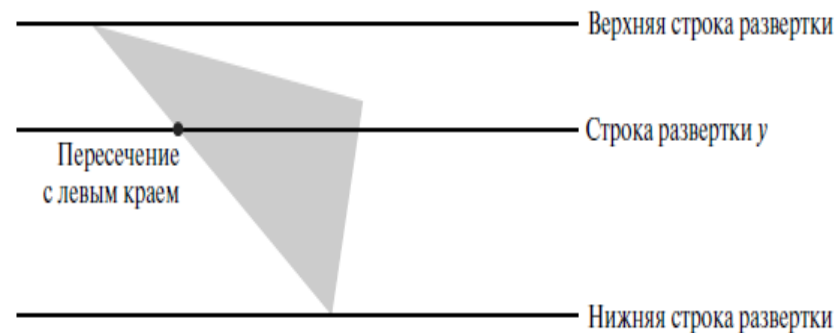


Рис. 1. Строки развертки, пересекающие многоугольную поверхность

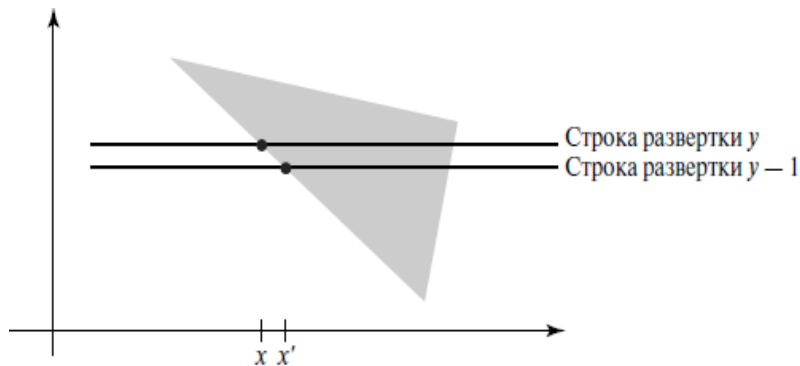


Рис. 2. Точки пересечения вдоль левого края многоугольника в последовательных строках развертки

Альтернативный подход – использовать для определения начальных значений x вдоль сторон для каждой строки развертки метод средней точки или разновидность алгоритма Брезенхема.

МЕТОД БУФЕРА ГЛУБИНЫ

- Плюсы
- Этот метод можно применить и к криволинейным поверхностям, определяя глубину и цвет в каждой точке проекции поверхности.
- Для многоугольных поверхностей метод буфера глубины легко реализовать, не сортируя поверхности сцены.
- Проблемы
- Требуется наличие второго буфера в дополнение к буферу регенерации. Система с разрешением 1280 на 1024, например, потребует свыше 1,3 миллионов позиций в буфере глубины, причем каждая позиция должна содержать достаточное число битов для представления необходимого разрешения по глубине.
- Кроме того, стандартный алгоритм буфера глубины часто выполняет ненужные вычисления. Объекты обрабатываются в произвольном порядке, может вычисляться цвет точки поверхности, которая позднее окажется невидимой.
- **Недостаток** метода буфера глубины заключается в том, что он определяет только одну видимую поверхность в каждом пикселе, работает только с непрозрачными поверхностями и не может накапливать коды цветов для нескольких поверхностей.

МЕТОД А-БУФЕРА

- Расширение метода буфера глубины. метод с защитой от наложения, усреднением по области и выявлением видимых поверхностей, разработанный Lucasfilm Studios для системы визуализации поверхностей REYES (Renders Everything You Ever Saw Визуализирует все, что вы видите).
- В области буфера (буфер накопления) хранятся различные данные о поверхности. Каждая позиция соответствует списку поверхностей, позволяет вычислить цвет пикселя как комбинацию цветов различных поверхностей при расчете прозрачности или защиты от наложения.
- Позиция имеет два поля.
 - Поле глубины - действительное значение (+, - или 0).
 - Поле данных о поверхности – данные о поверхности или указатель.
- Если поле глубины ≥ 0 , в этой позиции – глубина поверхности, проектирующейся в соответствующую область пикселя. Следующее поле - цвет поверхности для этой точки и процент охвата пикселя (рис. а). Если поле глубины для позиции в А-буфере < 0 , цвет пикселя определяется вкладами нескольких поверхностей.
- Следующее поле цвета - указатель на связанный список данных о поверхности, (рис., б).
- Поверхности делятся на многоугольную сетку и обрезаются по границам пикселей. Используя параметры прозрачности и процент охвата поверхности, алгоритмы визуализации рассчитывают цвет каждого пикселя как среднее вкладов перекрывающихся поверхностей.

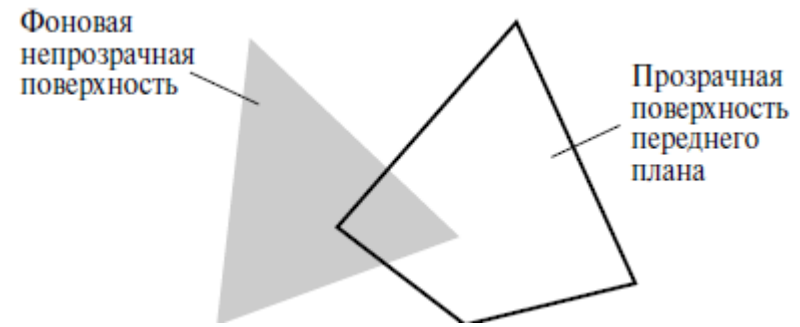


Рис. Для наблюдения непрозрачной поверхности через прозрачную следует ввести несколько цветов и смешать их

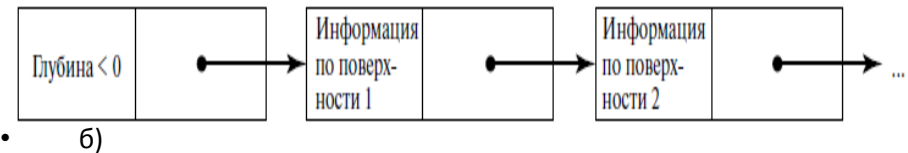
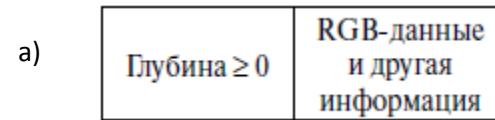


Рис. Два возможных представления информации о поверхности для пикселя в А-буфере. Если в пиксель проектируется одна поверхность, глубина, цвет и другая информация, записывается так (а). Если на пиксель накладывается несколько поверхностей, соответствующий список данных записывается так (б)

Информация о поверхности в А-буфере

- значения интенсивностей RGB-компонентов,
- параметр непрозрачности (процент прозрачности),
- глубина,
- процент охвата площади,
- идентификатор поверхности,
- другие параметры, требуемые при визуализации поверхности.

МЕТОД СТРОК РАЗВЕРТКИ

- Метод пространства изображений для выявления видимых компонентов поверхностей и сравнения глубин вдоль различных строк развертки на сцене.
1. При обработке строки развертки исследуются все проекции многоугольных поверхностей, пересекающих строку, и определяется, какая из них видима.
 2. Вдоль каждой строки развертки рассчитывается глубина, а по ней в каждом пикселе определяется, какая поверхность расположена ближе всего к плоскости наблюдения. Когда для пикселя определена видимая поверхность, ее цвет вводится в буфер кадров.
- Используют таблицы многоугольников. Координаты конечных точек всех линий на сцене, обратных наклонов каждой линии и указателей на таблицу граней поверхностей записаны в **таблице сторон**. **Таблица граней** содержит коэффициенты плоскости, свойства материалов поверхности, другие данные, и ссылки на таблицу сторон.
 - Для каждой строки развертки формируется активный список сторон – в порядке увеличения x записаны стороны, пересекающиеся с данной строкой развертки.
 - Для каждой поверхности определяется метка (flag), положение включено или выключено и указывает, где располагается точка строки развертки внутри или снаружи поверхности.
 - Положения пикселей вдоль каждой строки развертки обрабатываются слева направо.

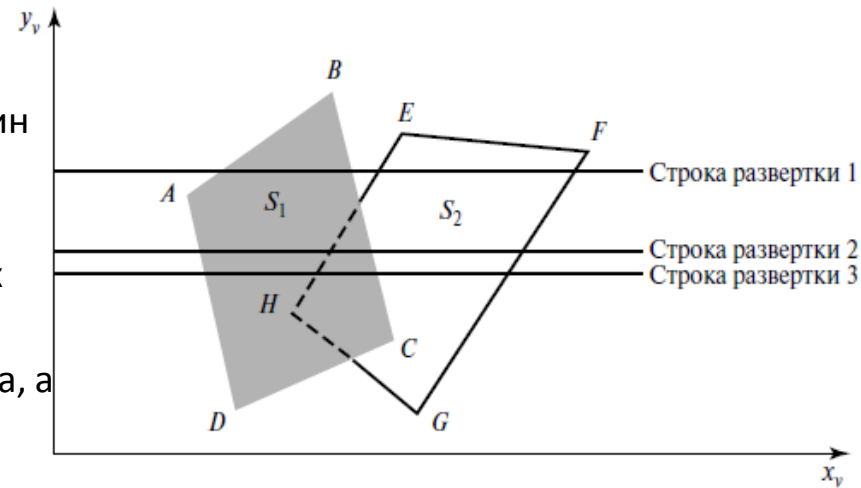
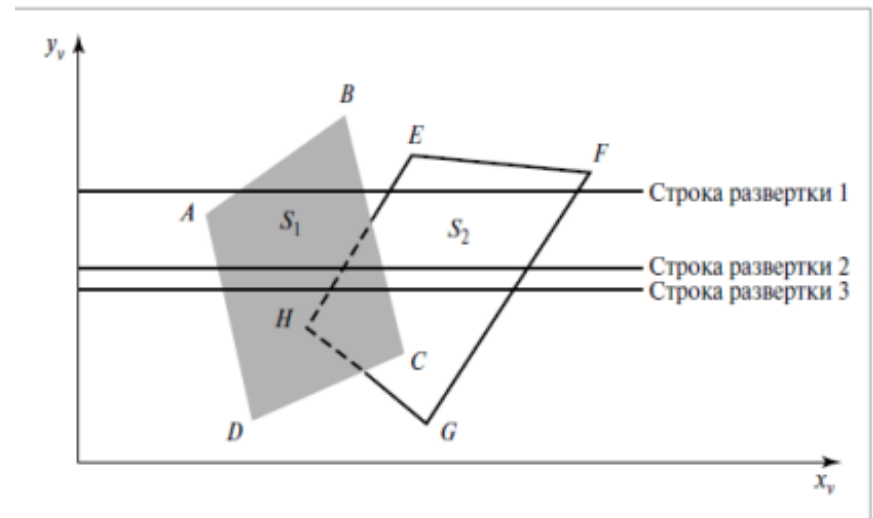


Рис.. Строки развертки, пересекающие проекцию на плоскость наблюдения двух поверхностей S_1 и S_2 . Пунктиром показаны границы невидимых участков поверхности

- В крайне слева точке пересечения с плоской проекцией выпуклого многоугольника метка поверхности включается,
- в крайней справа точке пересечения вдоль строки развертки она выключается.
- Для вогнутого многоугольника пересечения со строками развертки также можно сортировать слева направо, а метку поверхности включать между парой пересечений.

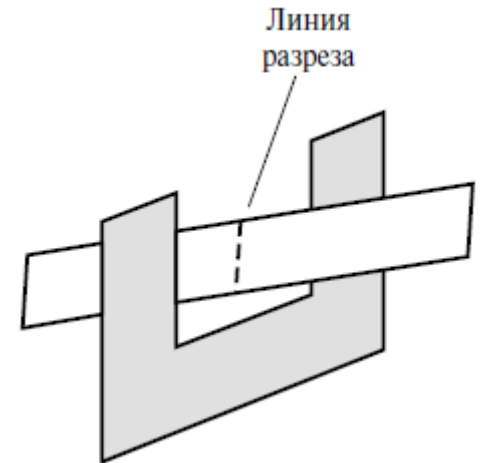
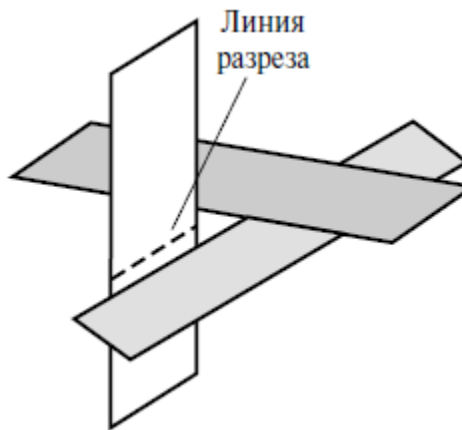
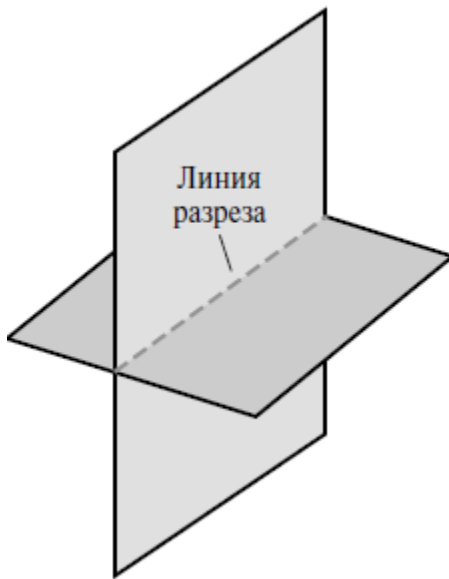
МЕТОД СТРОК РАЗВЕРТКИ

- Активный список для **строки развертки 1** содержит информацию из таблицы сторон, АВ, ВС, ЕН и FG.
- Вдоль этой строки между сторонами АВ и ВС – 1 поверхность S_1 . глубины вычислять не надо, коды цвета определяются по свойствам и условиям освещенности поверхности S_1 .
- Аналогично между ЕН и FG в состоянии включено находится метка поверхности S_2 . цвет пикселей совпадает с цветом фона, и его можно загрузить в буфер кадров при инициализации.
- Для строк развертки 2 и 3 на рис. активный список сторон содержит стороны AD, ЕН, ВС и FG. Вдоль строки 2 от стороны AD до стороны ЕН включена только метка поверхности S_1 .
- между сторонами ЕН и ВС включены метки обеих поверхностей. При обработке грани ЕН необходимо вычислить глубину, используя коэффициенты плоскостей двух поверхностей. В примере глубина поверхности S_1 меньше глубины поверхности S_2 . Следовательно, пикселям строки развертки присваивается цвет поверхности S_1 , до границы ВС.
- Затем метка поверхности S_1 выключается, и до края FG - цвета поверхности S_2 расчетов глубины не требуется, предполагается, что поверхность S_2 позади S_1 после того, как их связь по глубине определена при обработке стороны ЕН.
- При переходе от одной строки развертки к следующей используем когерентность строк развертки.
- строка развертки 3 имеет тот же активный список
- сторон, что и 2. относительно пересечений строк
- никаких изменений не наблюдается, то вычислять
- и сравнивать глубины сторон ЕН и ВС не надо.
- Две поверхности должны сохранять ориентацию,
- определенную в строке развертки 2, так что
- в буфер кадров можно без дополнительных
- расчетов глубины вводить цвета поверхности S_1 .



МЕТОД СТРОК РАЗВЕРТКИ

- С помощью описанного метода можно обработать любое число накладывающихся многоугольных поверхностей. Для поверхностей устанавливаются метки - точка внутри или снаружи поверхности, глубина рассчитывается только на краях накладывающихся поверхностей.
- Данная процедура корректно работает, когда поверхности не проходят одна сквозь другую или нет какого-либо иного циклического наложения (рис.). Если на сцене циклическое наложение, то поверхности можно разделить на части. Плоскости разрезаются на два отдельных участка (пунктир на рисунке), что позволяет избежать циклического наложения.



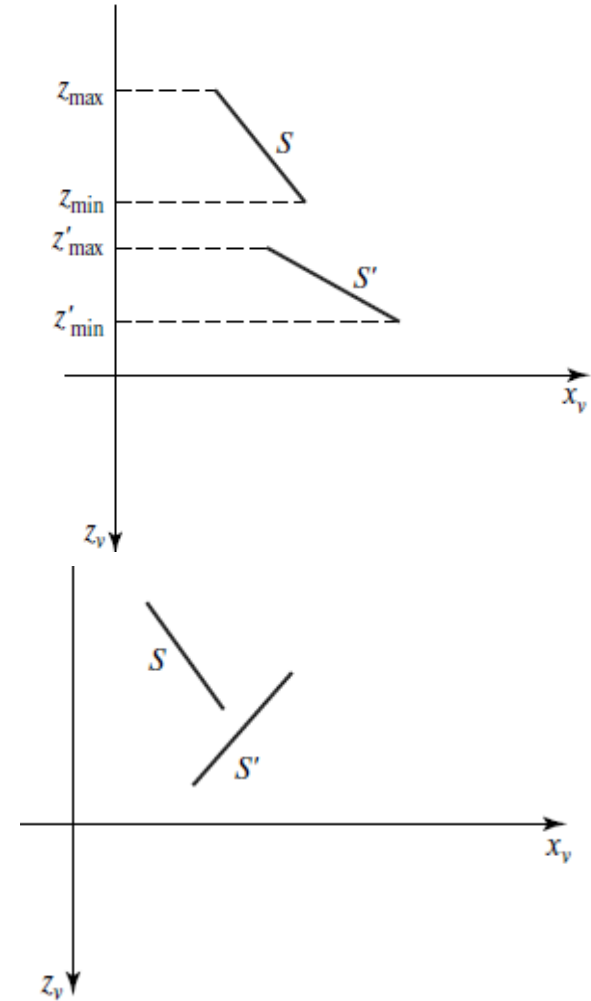
- а) б) в)
- Рис. 9.11. Пересечение и циклическое наложение поверхностей, попеременно затеняющих одна другую

МЕТОД СОРТИРОВКИ ПО ГЛУБИНЕ

- Метод сортировки по глубине (алгоритм художника). Художник вначале наносит фоновые цвета. Затем добавляются наиболее удаленные объекты, потом более близкие объекты и т.д. Каждый слой цвета закрывает предыдущий уровень. Реализует следующие базовые функции.
 - 1. Поверхности сортируются в порядке уменьшения глубины.
 - 2. Поверхности упорядочиваются согласно стандарту развертки, начиная с наиболее глубокой.
 - Операции сортировки выполняются и в пространстве изображений, и в пространстве объектов, а преобразование многоугольных поверхностей в стандарт развертки выполняется в пространстве изображений.
 - Поверхности вначале сортируются по расстоянию от плоскости наблюдения.
 - Затем в буфер регенерации вводятся коды цветов далекой поверхности. При обработке каждой последующей поверхности рисуют поверхность в буфере кадров поверх цветов ранее обработанных.
 - Упорядоченное рисование многоугольных поверхностей в буфере кадров выполняется в несколько этапов.
1. Сцена наблюдается по направлению оси z , при первом проходе поверхности упорядочиваются согласно наименьшему значению z каждой поверхности.
 2. Затем поверхность S в конце списка (с наибольшей глубиной) сравнивается с другими поверхностями списка, и определяется, перекрывается ли она с ними по глубине.
 3. Если нет, S наиболее удаленная, и она преобразовывается в стандарт развертки.
 4. Затем данный процесс повторяется для следующей поверхности списка.
 5. Если перекрытия нет, поверхности обрабатываются в порядке, определяемом глубиной, пока все они не будут преобразованы в стандарт развертки.
 6. Если для некоторой позиции списка обнаруживается перекрытие по глубине, необходимы дополнительные сравнения, чтобы выяснить, нужно ли переупорядочить какие-либо поверхности.

МЕТОД СОРТИРОВКИ ПО ГЛУБИНЕ

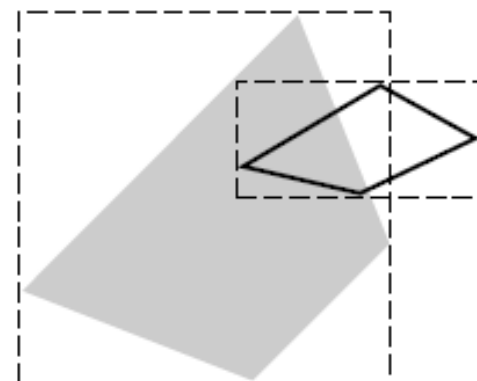
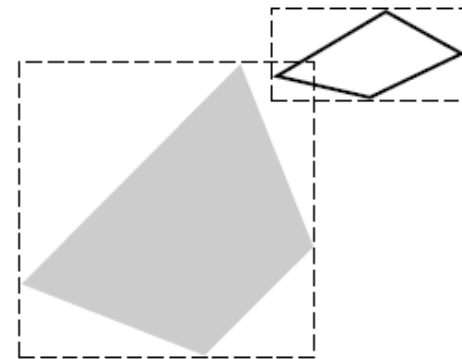
- В качестве примера на рис. 1 две поверхности, накладывающиеся на плоскости x_y , но не имеющие перекрытия по глубине.
- Для каждой поверхности, перекрывающейся по глубине с S , выполняются проверки. Если в любой из этих проверок получен положительный ответ, значит, переупорядочения выполнять не нужно, и поверхность S прошла проверку.
- Проверки в порядке увеличения сложности.
- 1. Ограничивающие прямоугольники (координатные границы) двух поверхностей в направлениях x и y не перекрываются.
- 2. Поверхность S полностью расположена за некоторой поверхностью относительно точки наблюдения.
- 3. Накладывающаяся поверхность полностью находится перед S относительно точки наблюдения.
- 4. Проекции ограничивающих областей двух поверхностей на плоскость наблюдения не перекрываются.
- проверки выполняются в указанном порядке, если какая-то проверка положительна, начинается обработка следующей перекрывающейся поверхности. Если все поверхности дали положительный ответ хотя бы при одной проверке, **S наиболее удаленная поверхность**.
- В таком случае переупорядочения не требуется, S преобразуется в стандарт развертки.



- Рис. 1. Две поверхности без наложения по глубине
- Рис. 2. Две поверхности с перекрытием по глубине, но без перекрытия по оси x

МЕТОД СОРТИРОВКИ ПО ГЛУБИНЕ

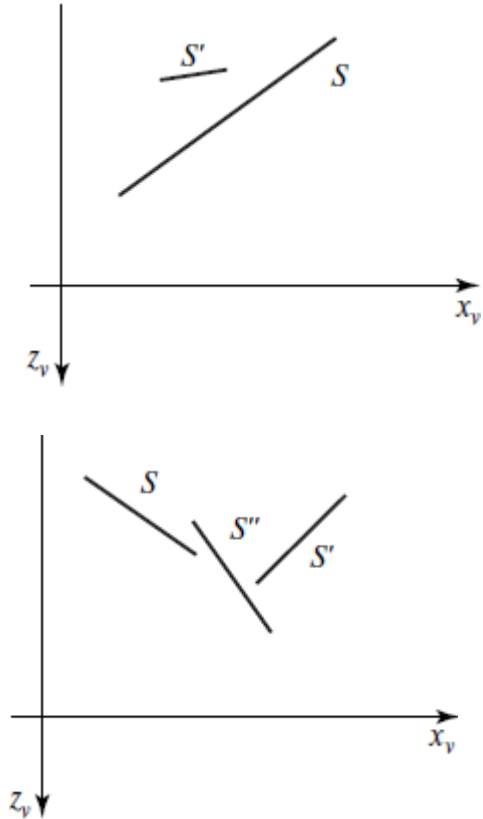
- Проверка 1 выполняется в два этапа. Вначале наложение по оси x , затем по y . Если ни в одном направлении нет перекрытия поверхностей, две плоскости не могут закрывать одна другую.
- Проверки 2 и 3 можно выполнить с использованием проверки передняя/задняя грань многоугольника.
- Координаты всех вершин S подставляются в уравнение плоскости для накладываемых поверхностей, и проверяется знак результата.
- Если уравнения плоскости заданы так, что передняя сторона поверхности смотрит на точку наблюдения, то S расположена за S' , если все вершины S находятся позади S' .
- Таким образом, S' полностью находится перед S , если все вершины S' расположены перед S .
- Если все проверки 1-3 дали отрицательный ответ, выполняется проверка 4, по результату которой определяется, налагаются ли две поверхности. Как показано на рис., две поверхности могут пересекаться или не пересекаться, даже когда их координатные границы налагаются.
- Если все четыре проверки дают отрицательные результаты для проектирующейся поверхности S' , поверхности S и S' в списке сортировки меняются местами.



- а) б)
- Рис.. Две многоугольные поверхности с накладываемыми граничными прямоугольниками на плоскости xy

МЕТОД СОРТИРОВКИ ПО ГЛУБИНЕ

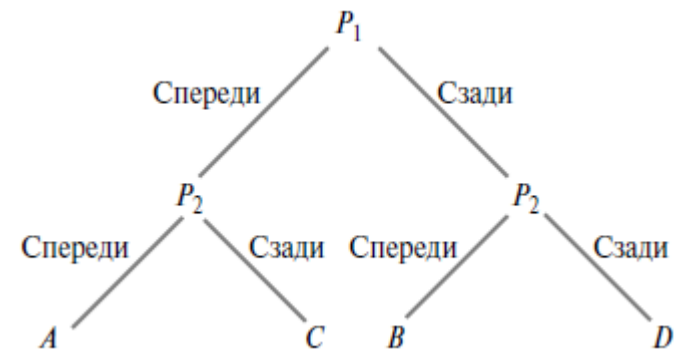
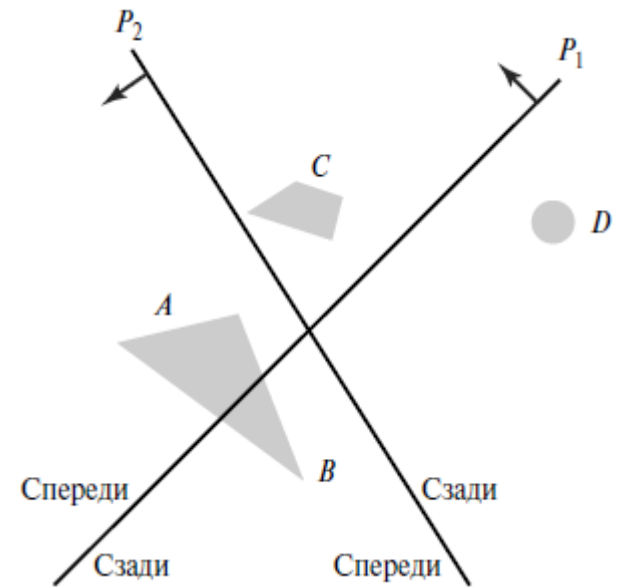
- Пример двух поверхностей, переупорядоченных с помощью описанной процедуры, (рис. 6.)
- На этом этапе еще достоверно неизвестно, найдена ли плоскость, наиболее удаленная от плоскости наблюдения.
- На рис. 7 ситуация, когда вначале меняются местами S и S'' .
- Однако, S'' скрывает часть S' , необходимо еще поменять местами S'' и S' , чтобы три поверхности шли в правильном порядке глубины.
- Следовательно, для каждой поверхности, порядок которой в списке меняется, ряд проверок нужно повторить.
- Если несколько поверхностей попеременно закрывают друг друга, алгоритм может просто зацикливаться.
- Чтобы избежать подобных циклов, можно пометить любую поверхность, передвинутую глубже, чтобы ее нельзя было снова переместить.
- При попытке переставить поверхность повторно, она делится на две части, исходная поверхность заменяется двумя новыми, и обработка продолжается, как раньше



- Рис. 6. Некоторые точки поверхности S имеет большую глубину, чем все точки поверхности S' , но поверхность S затеняет поверхность S'
- Рис. 7. Три поверхности введены в список сортировки в порядке S, S', S'' , но они должны упорядочиться как S', S'', S

МЕТОД BSP-ДЕРЕВА

- Метод дерева бинарного разбиения пространства (binary space-partitioning BSP) позволяет определить видимость объекта, занося поверхности в буфер кадров от фона к переднему плану.
- Применение BSP-дерева в проверках видимости включает выявление на каждом этапе деления пространства поверхностей, расположенных за или перед плоскостью разбиения (относительно направления наблюдения). (рис.)
- Для плоскости P_1 пространство вначале делится на 2 набора объектов. Один позади P_1 относительно направления наблюдения, другой перед P_1 . плоскость P_1 пересекает объект, он делится на два разных объекта, A и B . Объекты A и C перед P_1 , а B и D за P_1 .
- Далее пространство делится плоскостью P_2 , строится представление в форме двоичного дерева (рис., б). объекты представляются узлами, передние объекты занимают левые ветви, а задние объекты соответствуют правым ветвям.
- Для объектов, описанных многоугольными гранями, плоскости деления выбирают совпадающими с плоскостями поверхностей многоугольников.
- с помощью уравнений многоугольников определяются задние и передние многоугольники, и дерево строится с одной плоскостью деления для каждой грани.
- многоугольник, пересекаемый плоскостью деления, разбивается на две части. BSP-дерево обрабатывается в следующем порядке. вначале правые узлы, затем левые. поверхности подготавливаются к отображению от фона к переднему плану, объекты переднего плана рисуются поверх фоновых.



- а) б)
- Рис. Область пространства (а) делится на две плоскости P_1 и P_2 с целью получения BSP-дерева (б)
- .

МЕТОДЫ ДЕЛЕНИЯ ОБЛАСТИ

- Данный метод удаления невидимых поверхностей это, по сути, метод пространства изображений, но для упорядочения поверхностей по глубине могут использоваться ориентации в пространстве объектов.
- Метод состоит в делении всей области плоскости наблюдения на последовательно уменьшающиеся прямоугольники.
- Нужно установить проверки - область является частью одной поверхности, или что эта область слишком сложна, чтобы ее можно было просто проанализировать.
- Изначально к общей проекции применяются проверки, позволяющие определить, нужно ли разделить общую область на меньшие прямоугольники. Если проверки указывают, что проекция достаточно сложна, она делится на части.
- Затем проверки применяются ко всем меньшим областям, которые при необходимости также разбиваются (если проверки показали, что видимость суммарной поверхности все еще неопределенная).
- Процесс продолжается, пока не удастся однозначно отнести фрагмент к одной поверхности, или пока не будет достигнуто предельное разрешение.
- На каждом шаге нужно последовательно делить область на четыре равные части (рис.) Данный подход подобен используемому при построении квадродерева (четверичного дерева, quadtree)..

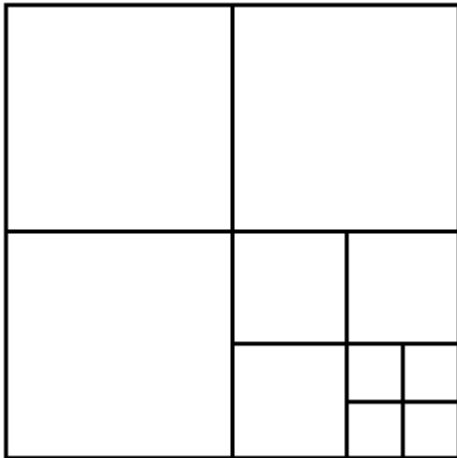
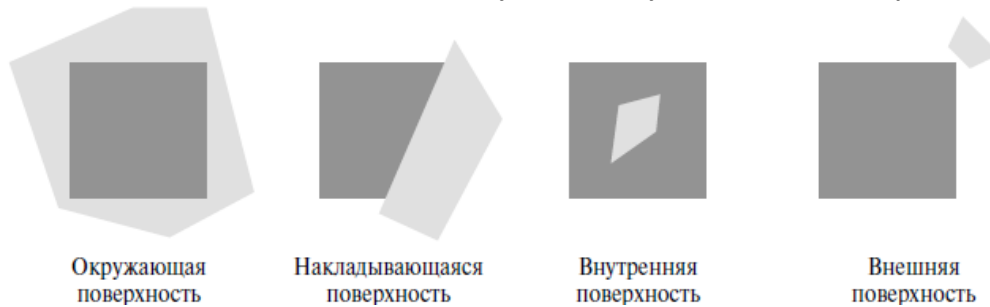


Рис.. Деление квадратной области на каждом шаге на квадраты равного размера

МЕТОДЫ ДЕЛЕНИЯ ОБЛАСТИ

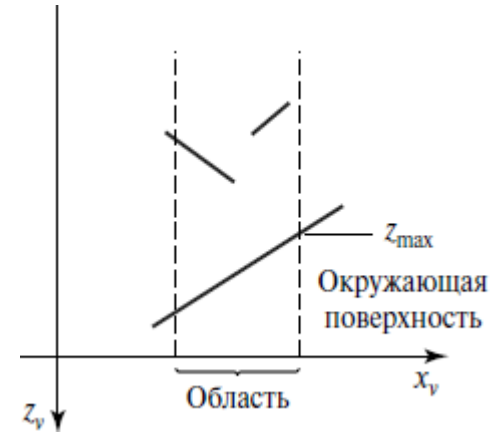
- Существует четыре возможные связи, которые поверхность может иметь с областью разделенной плоскости наблюдения. Используют следующие классификации (рис.).



- Рис. Возможные связи между многоугольными поверхностями и прямоугольной областью плоскости наблюдения
- Окружающая поверхность:** поверхность, полностью замыкающая область.
- Накладывающаяся поверхность:** поверхность, расположенная частично внутри и снаружи области.
- Внутренняя поверхность:** поверхность, расположенная целиком внутри области.
- Внешняя поверхность:** поверхность, расположенная полностью вне области.
- Проверки на определение видимости поверхности в пределах прямоугольной области основаны на 4-ех классификаций поверхности (рис.). Дальнейшее деление указанной области не требуется, если выполняется одно из условий.
- Условие 1. область не принадлежит к внутренним, накладывающимся или окружающим поверхностям (все поверхности находятся вне области).
- Условие 2. область имеет только одну внутреннюю, накладывающуюся или окружающую поверхность.
- Условие 3. область имеет одну окружающую поверхность, затеняющую все остальные поверхности в пределах границ области

МЕТОДЫ ДЕЛЕНИЯ ОБЛАСТИ

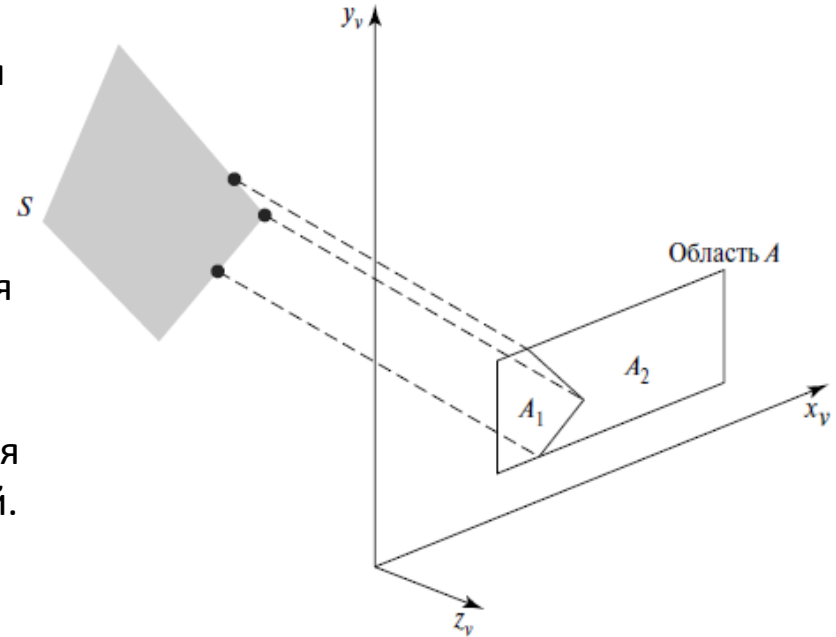
- Первый шаг – сравнение координатных границ всех поверхностей в пределах границ области. Определяют внутренние и окружающие поверхности
- Выявление накладывающихся и внешних поверхностей требует проверок на пересечение. Если первый граничный прямоугольник пересекает область, то, чтобы определить, является поверхность окружающей, накладывающейся или внешней, используются дополнительные проверки.
- Как только выявлена внутренняя, накладывающаяся или окружающая поверхность, в буфер кадров записываются ее коды цвета.
- Чтобы проверить условие 3, можно, рассортировать поверхности по минимальной глубине от плоскости наблюдения. Затем для каждой окружающей поверхности вычисляется максимальная глубина в пределах области.
- Если максимальная глубина (отсчитывается от плоскости наблюдения) одной из окружающих поверхностей меньше, чем минимальная глубина всех остальных поверхностей, условие 3 удовлетворяется (рис.).
- Другой метод проверки условия 3 – использовать уравнения плоскости для расчета значений глубин в 4-ех вершинах области для всех окружающих, накладывающихся и внутренних поверхностей. Если все четыре глубины для одной поверхности меньше рассчитанных глубин для всех остальных поверхностей, условие 3 удовлетворено. Далее область можно отобразить с цветами окружающей поверхности.



- Рис.. В заданной области окружающая поверхность с максимальной глубиной z_{\max} затеняет все поверхности, минимальная глубина которых превышает z_{\max}

МЕТОДЫ ДЕЛЕНИЯ ОБЛАСТИ

- Область проще разделить. Поверхность, классифицированная как внешняя или окружающая для данной области, будет оставаться такой при делениях области.
- Когда деление доходит до размера пикселя, в каждой точке рассчитывается глубина значимой для нее поверхности, и пикселю присваивается цвет ближайшей поверхности.
- Возможна модификация базового процесса деления области - можно делить вдоль границ поверхностей.
- Если поверхности были отсортированы по минимальной глубине, для деления используется поверхность с наименьшим значением глубины.
- рис.
- Проекция границы поверхности S делит исходную область на подобласти A_1 и A_2 .
- После этого поверхность S является окружающей поверхностью для A_1 , и, чтобы определить, требуется ли дальнейшее деление, можно проверять условия видимости 2 и 3.



- Рис. Область A делится на A_1 и A_2 с помощью границы поверхности S на плоскости наблюдения

МЕТОДЫ ОКТОДЕРЕВЬЕВ

- Выявление видимых поверхностей через поиск в пространстве узлов октодерева от переднего к заднему плану. На рис. передний план сцены - октанты 0, 1, 2 и 3. Поверхности перед ними видимы для наблюдателя. Любые поверхности, позади передних октантов или в задних октантах (4, 5, 6 и 7), могут быть скрыты передними поверхностями.
- Узлы октодерева обработать в порядке 0, 1, 2, 3, 4, 5, 6, 7.
- Соответствует прохождению дерева вглубь, узлы четырех передних подоктантов октанта 0 видимы перед узлами четырех задних подоктантов. При каждом делении октанта прохождение октодерева выполняется в таком порядке.
- Когда при проходе восьмеричного дерева встречается код цвета, цвет записывается в квадродереве, только если для этой области ранее не было записано ни одного значения.
- Записываются только цвета передних граней. Узлы, имеющие значение пусто(void), игнорируются.
- Любой полностью затеняемый узел исключается из дальнейшей обработки, так что его поддеревья недоступны.
- На рис. октанты области пространства и соответствующие квадранты на плоскости наблюдения. Вклады в квадрант 0 дают октанты 0 и 4. Коды цвета в квадранте 1 определяют поверхности в октантах 1 и 5, а коды в остальных двух квадрантах генерируются по значениям в парах октантов, соотнесенных с этими квадрантами.
- Эффективно проверить видимость по октодереву позволяет рекурсивная обработка узлов октодерева и создание квадродерева для видимых поверхностей.

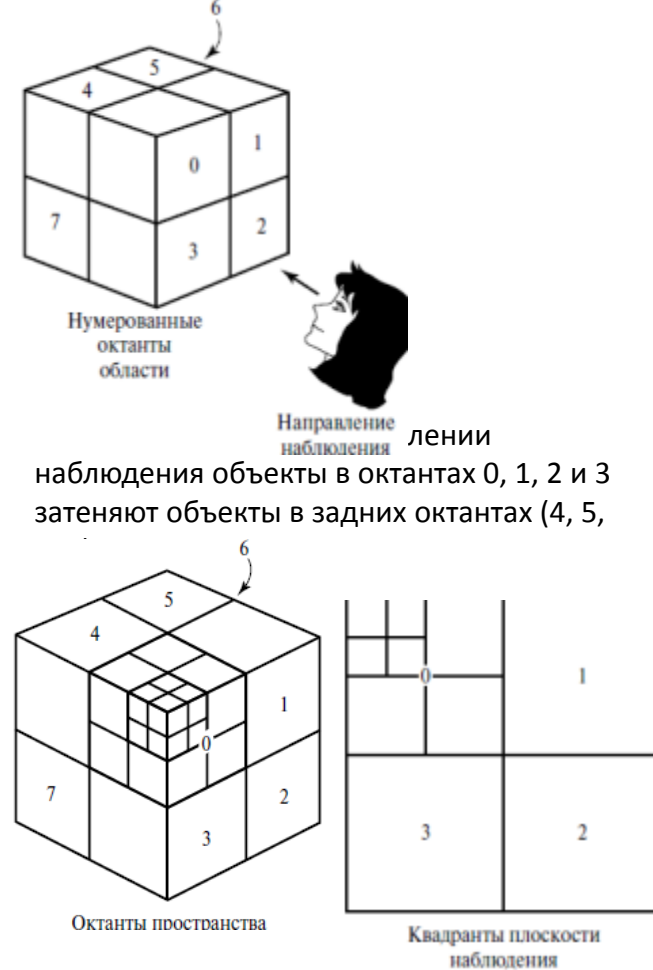
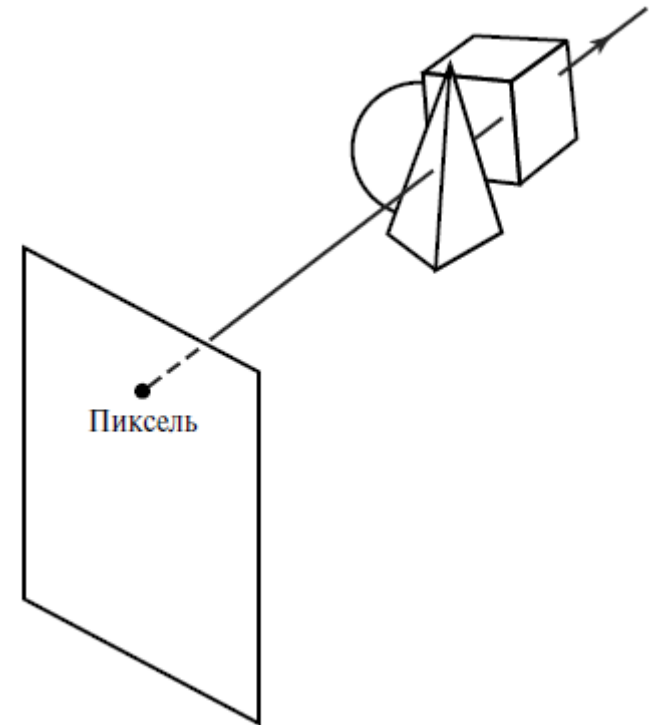


Рис. Этапы деления октантов области пространства и плоскость соответствующего квадранта

МЕТОД РАСЧЕТА ЛУЧА

- Если рассмотреть луч, идущий от положения пикселя на плоскости наблюдения через сцену, рис., можно определить, какие объекты пересекают луч. Видимой считается поверхность, точка пересечения с которой является ближайшей к пикселю.
- Расчет луча основан на методах геометрической оптики, позволяющих отследить пути лучей света. можно отследить ход лучей света в обратном порядке от пикселей на сцену.
- Метод расчета лучей при анализе видимых поверхностей эффективен для сцен с криволинейными поверхностями, сферами.
- В методе расчета лучей по одному обрабатываются пиксели, и для всех поверхностей, проектирующихся в этот пиксель, вычисляются глубины.
- Расчет луча является частным случаем алгоритма построения хода луча (raytracing)), прокладывающим несколько путей луча, чтобы учесть глобальные вклады в отражение и преломление от нескольких объектов сцены. При расчете же луча луч прослеживается только от каждого пикселя до ближайшего объекта.



- Рис.. Луч вдоль линии наблюдения от пикселя сквозь сцену

СРАВНЕНИЕ МЕТОДОВ ИССЛЕДОВАНИЯ ВИДИМЫХ ПОВЕРХНОСТЕЙ

- Эффективность метода исследования видимых поверхностей зависит от характеристик конкретного приложения. Если поверхности на сцене значительно распределены вдоль направления наблюдения, и существует крайне незначительное перекрытие по глубине, часто эффективнее использовать методы сортировки по глубине или BSP-дерева. Если проекции поверхностей на плоскость наблюдения немного перекрываются, видимые поверхности проще локализовать с использованием строк развертки или деления области.
- Можно сформулировать общее правило. если сцена содержит малое число поверхностей, крайне эффективным подходом является алгоритм сортировки по глубине или метод BSP-дерева. Это объясняется тем, что подобные сцены обычно содержат мало поверхностей, перекрывающихся по глубине.
- Если сцена содержит криволинейные поверхности, то, чтобы определить видимые части сцены, можно использовать метод октодерева или расчета лучей.
- Когда в системе используется октодерево, процесс исследования видимых поверхностей быстр и прост. В нем используются только целые сложения и вычитания, поэтому не нужно выполнять сортировку или расчет точек пересечения. Другое достоинство октодеревьев заключается в том, что в них записана не только геометрия поверхностей. Для отображения доступен весь объем объекта, что делает представление в форме октодеревьев полезным при расчете сечений трехмерных объектов.
- Методы исследования видимых поверхностей можно различным образом применять и объединять.

КРИВОЛИНЕЙНЫЕ ПОВЕРХНОСТИ

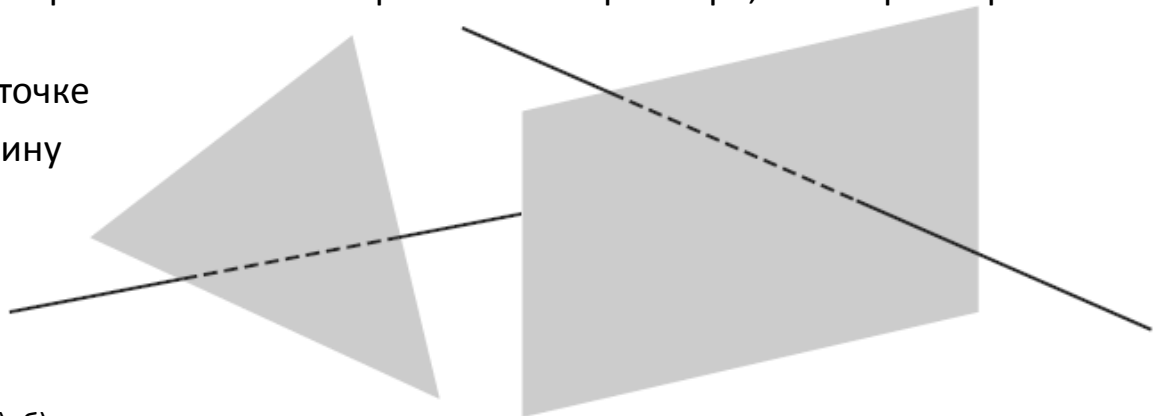
- В число эффективных методов входят расчет лучей и метод октодеревя. При расчете лучей определяются точки пересечения луча с поверхностями и расстояние вдоль луча до ближайшей точки пересечения. При использовании октодеревьев прослеживаются узлы от переднего к заднему плану, и определяются коды цветов поверхностей. Все видимые поверхности выявляются с помощью рассмотренных процедур обработки. Никаких принципиальных отличий при работе с криволинейными поверхностями не возникает.
- Криволинейную поверхность можно также аппроксимировать многоугольной сеткой, а затем использовать один из методов исследования видимых поверхностей. Для некоторых объектов (например, сфер) эффективнее и точнее будет использовать расчет лучей и уравнения криволинейных поверхностей.
- Поверхность можно представить с помощью неявного уравнения вида $f(x, y, z)=0$ или посредством параметрического представления. Сплайновые поверхности, например, обычно описываются параметрическими уравнениями.
- В некоторых случаях бывает полезно получить и явное уравнение поверхности, например, уравнение высот точек над плоскостью xy .
- $z = f(x, y)$.
- сферы, эллипсоиды, цилиндры и конусы, имеют квадратическое представление и широко используются для моделирования.
- В алгоритмах строк развертки и расчета луча часто фигурируют методы численной аппроксимации, с помощью которых решается уравнение поверхностей в точке пересечения со строкой развертки или лучом пикселя.

ПРЕДСТАВЛЕНИЯ КРИВОЛИНЕЙНЫХ ПОВЕРХНОСТЕЙ

- ГРАФИКИ КОНТУРОВ ПОВЕРХНОСТИ
- Поверхность можно описать уравнением или таблицами данных. Имея явное функциональное представление, можно изобразить видимые контурные линии поверхности и удалить участки контура, скрытые видимыми частями поверхности. Чтобы получить график в координатах функциональной поверхности xu , представление поверхности записать в виде
- $y = f(x, z)$.
- Затем кривая на плоскости xu изображается для z , принадлежащих выбранному диапазону при заданном интервале z . Начиная с наибольшего z , рисуются кривые от переднего к заднему плану, и удаляются скрытые сегменты. Сегменты кривых представляются на экране как отображение диапазона xu функции в диапазон xu пикселей экрана. Затем по x берутся 1-ые шаги, и для каждого x при данном значении z определяется соответствующее значение y .
- Способ определения видимых сегментов кривых на поверхности - поддерживается список значений y_{\min} и y_{\max} , ранее рассчитанных для координат x пикселя на экране. При переходе от одного положения x к следующему значение y проверяется на принадлежность диапазону y_{\min} и y_{\max} для следующего пикселя. Если $y_{\min} \leq y \leq y_{\max}$, то точка невидима, не изображается. Если значение y вне границ для пикселя, точка видима. рисуется точка и обновляются границы этого пикселя. Подобные же процедуры для xz или yz .
- Те же методы и к дискретному набору точек данных для определения горизонталей. Например, дан дискретный набор значений z для сетки px на py значений xu , на поверхности можно определить путь линии постоянного z , используя методы построения контуров. Затем каждую выбранную линию контура можно спроектировать на плоскость наблюдения и отобразить прямыми отрезками. Как и ранее, линии можно рисовать в порядке от переднего плана к заднему и не изображать части контуров, проходящие за ранее нарисованными (видимыми) линиями контуров.

МЕТОДЫ ОПРЕДЕЛЕНИЯ ВИДИМОСТИ ДЛЯ КАРКАСНЫХ ИЗОБРАЖЕНИЙ

- Наиболее быстрый способ получения каркасного вида - отобразить все грани объектов, но трудно определить передние и задние элементы. Решение – в применении затухания с глубиной (depth cueing), интенсивность линии - функция ее расстояния от наблюдателя. Альтернативно можно использовать проверки видимости, и, либо удалить скрытые отрезки, либо изобразить их иначе (пунктир), чем видимые края.
- АЛГОРИТМЫ ИССЛЕДОВАНИЯ ВИДИМЫХ ПОВЕРХНОСТЕЙ ДЛЯ КАРКАСНЫХ ОБЪЕКТОВ
- Прямой подход – сравнение положений краев с положениями поверхностей на сцене. Используются те же методы, что и в алгоритмах отсечения линий. Положение конечных точек отрезка проверяется относительно границ области, при проверке на видимость сравнить еще **глубины краев и поверхности**.
 1. Если обе конечные точки отрезка принадлежат области проекции поверхности, глубина конечных точек сравнивается с глубиной поверхности (x, y) в этих точках.
 2. Если обе конечные точки находятся за поверхностью, значит, край скрыт.
 3. Если обе конечные точки находятся перед поверхностью, край является видимым.
- В противном случае рассчитать точки пересечения и определить глубины в них.
 1. край имеет большую глубину, чем поверхность в точках пересечения периметра, часть края скрыта. рис., а.
 2. край имеет большую глубину в 1-ой точке пересечения с границей и меньшую глубину в другой (поверхности выпуклые).
Определить, где край проникает внутрь, рис., б. скрытую часть края удалить, изобразить пунктиром т.д..



а) б)

Рис. Скрытые части линии (пунктир), имеющей большую глубину, чем поверхность (а), и расположенной частично за и перед поверхностью (б)

МЕТОДЫ ОПРЕДЕЛЕНИЯ ВИДИМОСТИ ДЛЯ КАРКАСНЫХ ИЗОБРАЖЕНИЙ

- Некоторые методы исследования видимых поверхностей легко приспособляются к проверке видимости каркаса из краев объекта. Используя метод задних граней, можно определить все задние поверхности объекта и отобразить только границы видимых поверхностей. Используя сортировку по глубине, поверхности можно нарисовать в буфере регенерации так, чтобы внутренние части поверхности имели цвет фона, а границы цвет переднего плана. Если обрабатывать поверхности от заднего плана к переднему, невидимые линии будут закрываться более близкими поверхностями. Алгоритм удаления невидимых линий можно дополнить методом деления площади, отображая только границы видимых поверхностей. Наконец, для отображения точек пересечения со строками развертки на границах видимых поверхностей можно использовать методы строк развертки.
- АЛГОРИТМ ЗАТУХАНИЯ С ГЛУБИНОЙ ДЛЯ КАРКАСНЫХ ОБЪЕКТОВ
- Другой метод отображения информации о видимости - изменение яркости объектов на сцене согласно расстоянию от точки наблюдения. Данный метод затухания с глубиной (depth-cueing method) обычно применяется с использованием линейной функции
- $$f_{\text{depth}}(d) = (d_{\text{max}} - d) / (d_{\text{max}} - d_{\text{min}})$$
- где d расстояние точки от наблюдателя. Значения минимальной и максимальной глубин d_{min} и d_{max} можно задать из соображений удобства. Или же минимальную и максимальную глубины можно задать в нормированном диапазоне частот. $d_{\text{min}} = 0, 0$ и $d_{\text{max}} = 1, 0$. При обработке каждого пикселя его цвет умножается на $f_{\text{depth}}(d)$. ближние точки отображаются с большими интенсивностями, на максимальной глубине – интенсивность 0.
- Функцию затухания с глубиной можно реализовать с различными вариациями. В некоторых графических библиотеках доступна общая атмосферная функция, которая позволяет объединять затухание с глубиной с атмосферными эффектами, например, для имитации дыма или тумана. Цвет объекта можно модифицировать функцией затухания с глубиной, а затем объединить с цветом атмосферы.

РЕЗЮМЕ

- Простейшей проверкой видимости - алгоритм исследования задних граней, быстрый и эффективный для начального отсеивания, устраняющего из дальнейших проверок на видимость значительное число многоугольников. Для одного выпуклого многогранника выявление задних граней исключает все невидимые поверхности, но в общем случае выявление задних граней не позволяет полностью найти все невидимые поверхности.
- Распространенным методом определения всех видимых поверхностей на сцене является алгоритм буфера глубины. При применении к стандартным графическим объектам данная процедура является весьма эффективной, но с ее использованием связаны дополнительные требования к памяти. Для работы метода нужно два буфера: один для хранения цветов пикселей и один для хранения значений глубины для положений пикселей. Для обработки каждого многоугольника на сцене (расчета глубин поверхностей) используются быстрые последовательные методы строк развертки. При обработке каждой последующей поверхности два буфера обновляются.
- Расширением подхода буфера глубины является метод A-буфера, который предоставляет дополнительную информацию, необходимую для защиты от наложения и отображения прозрачных поверхностей.
- Разработано еще несколько методов исследования видимых поверхностей. Метод строк развертки одновременно обрабатывает все поверхности для каждой строки развертки. В методе сортировки по глубине (алгоритм художника) объекты рисуются в буфере регенерации согласно их расстояниям от точки наблюдения. Определение видимых частей сцены с помощью схем деления представлено методом BSP-дерева, делением области и представлением в форме октодерева. Видимые поверхности также можно выявлять, используя методы расчета лучей, в которых путем проектирования линий с плоскости пикселей на сцену определяются точки пересечения этих линий проекции с объектами. Методы расчета лучей являются неотъемлемой частью алгоритмов построения хода луча, которые позволяют отображать сцены с эффектами глобального освещения.
- Методы исследования видимых поверхностей также используются при отображении трехмерных рисунков от руки (line drawings). При наличии на сцене криволинейных поверхностей можно отображать графики в горизонталях. Для построения каркасных изображений многогранников на сцене ищутся различные сегменты краев поверхностей, видимые из точки наблюдения.