

Путь: [Графика](#) »

Алгоритм Робертса может быть с успехом применен для изображения множества выпуклых многогранников на одной сцене в виде проволочной модели с удаленными невидимыми линиями. Метод непригоден непосредственно для передачи падающих теней и других сложных визуальных эффектов. Рассмотрим основные идеи метода на примере одиночного выпуклого многогранника.

Модель объекта

Объект-многогранник, состоящий из N граней, описывается матрицей T размера $4 \times N$, каждый n -й столбец которой содержит описание n -й ($n=[1,N]$) плоскости $A_nX + B_nY + C_nZ + D_n = 0$ в объектной системе координат XYZ :

$$T = \begin{vmatrix} A_1 & \dots & A_n & \dots & A_N \\ B_1 & \dots & B_n & \dots & B_N \\ C_1 & \dots & C_n & \dots & C_N \\ D_1 & \dots & D_n & \dots & D_N \end{vmatrix}$$

Минимальное число граней $K=4$ наблюдается у тетраэдра.

Важное требование к модели объекта в алгоритме Робертса заключается в достижении такой формы (знака) в функции каждой плоскости $f_n(X,Y,Z) = A_nX + B_nY + C_nZ + D_n$, при которой справедливо $f_n(X_{вн}, Y_{вн}, Z_{вн}) \geq 0$ для любой точки $(X_{вн}, Y_{вн}, Z_{вн})$, заведомо принадлежащей телу многогранника.

Достижение этого условия осуществляется путем опытной проверки знака функции относительно внутренней точки, в качестве которой может выступать точка со средним геометрическим положением относительно всех вершин многогранника. Формально правило формирования "положительной" грани может быть представлено в виде

$$f_{n+}(X,Y,Z) = f_n(X,Y,Z) \cdot \text{sign } f_n(X_{вн}, Y_{вн}, Z_{вн}),$$

где f_n - функция, описывающая n -ю грань с априори неизвестным знаком относительно внутренней точки; f_{n+} - та же функция, но с положительным значением. При достижении "положительности" всех граней автоматически достигается ориентация нормали

$$N_n = iA_n + jB_n + kC_n$$

к любой из граней внутрь фигуры.

Кроме параметров уравнений граней необходимо знать координаты вершин каждой грани. Вершины могут быть заданы или вычислены из матрицы T путем определения общих решений каждой плоскости со всеми остальными. В любом случае для каждой n -й грани должна быть составлена матрица K -числа вершин:

$$\begin{vmatrix} X_1 & Y_1 & Z_1 & 1 \end{vmatrix}$$

$$V_n = \begin{vmatrix} : & : & : & : \\ X_k & Y_k & Z_k & 1 \end{vmatrix}$$

Синтез изображения

Не все грани, входящие в состав объекта, будут видны наблюдателю. Невидимой будет та, нормаль N к которой ориентирована в ту же сторону, что и вектор R , направленный от объекта в сторону наблюдателя, т. е. при выполнении условия неотрицательности скалярного произведения $NR > 0$.

$$\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ : & : & : & : \\ x_k & y_k & z_k & 1 \end{vmatrix} = V_n M$$

Область внутри такой фигуры закрашивается соответствующим значением интенсивности, для вычисления которой и определения факта затененности телом самого многогранника вычисляют скалярное произведение нормального вектора N и вектора S , нацеленного из объекта на источник света. Греть освещена при отрицательности скалярного произведения, а величина освещенности выбирается пропорциональной значению $\cos \alpha = -S \cdot N / (|S| |N|)$. Соответственно грань находится в тени при условии $SN > 0$. Таким образом, алгоритм Робертса очень простыми действиями достигает успешного построения изображений отдельных выпуклых многогранных объектов.

Основным недостатком метода, определившим ограниченность его распространения, являются неспособность без привлечения других подходов реализовать падающие тени, невозможность передачи зеркальных эффектов и преломления света и, наконец, строгая ориентация метода только на выпуклые многогранники.

В модификациях алгоритма Робертса, ориентированных на изображение множества выпуклых объектов. Алгоритм выполняется в три этапа. На первом этапе алгоритм Робертса используется в классическом виде для удаления нелицевых (невидимых) граней каждого объекта в отдельности. На втором этапе каждое из множества оставшихся ребер оценивается на экранирование всеми другими телами. Наконец, на третьем этапе определяют отрезки, которые образуют новые ребра при протыкании телами друг друга. Несложно развить этот подход для закрашивания видимых граней, однако проблема падающих теней остается открытой.

Ниже приводится эффективная реализация алгоритма Робертса.

Этот алгоритм делится на три этапа. На первом этапе каждое тело анализируется индивидуально с целью удаления нелицевых плоскостей. На втором этапе проверяется экранирование оставшихся в каждом теле ребер всеми другими телами с целью обнаружения их невидимых отрезков. На третьем этапе вычисляются отрезки, которые образуют новые ребра при протыкании телами друг друга. В данном алгоритме предполагается, что тела состоят из плоских полигональных граней, которые в свою очередь состоят из ребер, а ребра - из отдельных вершин. Все вершины, ребра и грани связаны с конкретным телом.

Удаление нелицевых плоскостей

Для каждого тела в сцене:

Сформировать многоугольники граней и ребра, исходя из списка вершин тела.

Вычислить уравнение плоскости для каждой полигональной грани тела.

Проверить знак уравнения плоскости:

Взять любую точку внутри тела, например усреднив координаты его вершин.

Вычислить скалярное произведение уравнения плоскости и точки внутри тела.

Если это скалярное произведение < 0 , то изменить знак уравнения этой плоскости.

Сформировать матрицу тела.

Умножить ее слева на матрицу, обратную матрице видового преобразования, включающего перспективу.

Вычислить и запомнить габариты прямоугольной объемлющей оболочки преобразованного объема: X_{\max} , X_{\min} , Y_{\max} , Y_{\min} .

Определить нелицевые плоскости:

Вычислить скалярное произведение пробной точки, лежащей в бесконечности, на преобразованную матрицу тела.

Если это скалярное произведение < 0 , то плоскость невидима.

Удалить весь многоугольник, лежащий в этой плоскости. Это избавляет от необходимости отдельно рассматривать невидимые линии, образуемые пересечением пар невидимых плоскостей.

Удаление из каждого тела тех ребер, которые экранируются всеми остальными телами в сцене:

Если задано только одно тело, то алгоритм завершается.

Сформировать приоритетный список этих тел:

Провести сортировку по z . Сортировка производится по максимальным значениям координаты z , вершин преобразованных тел. Первым в упорядоченном списке и обладающим наибольшим приоритетом будет то тело, у которого минимальное среди максимальных значений z . В используемой правой системе координат это тело будет самым удаленным от точки наблюдения, расположенной в бесконечности на оси z .

Для каждого тела из приоритетного списка:

Проверить экранирование всех лицевых ребер всеми другими телами сцены. Тело, ребра которого проверяются, называется пробным объектом, а тело, относительно которого в настоящий момент производится проверка, называется пробным телом. Естественно, что нужно проверять экранирование пробного объекта только теми пробными телами, у которых ниже приоритеты.

Провести проверки экранирования для прямоугольных объемлющих оболочек пробного объекта и пробного тела:

Если $X_{\min}(\text{пробное тело}) > X_{\max}(\text{пробный объект})$ или $X_{\max}(\text{пробное тело}) < X_{\min}(\text{пробный объект})$ или $Y_{\min}(\text{пробное тело}) > Y_{\max}(\text{пробный объект})$ или $Y_{\max}(\text{пробное тело}) < Y_{\min}(\text{пробный объект})$, то пробное тело не может экранировать ни одного ребра пробного объекта. Перейти к следующему пробному телу. В противном случае:

Провести предварительные проверки протыкания, чтобы увидеть, не протыкается ли пробное тело пробным объектом и существует ли возможность частичного экранирования первого последним.

Сравнить максимальное значение z у пробного объекта с минимальным значением z у пробного тела.

Если $Z_{\max}(\text{пробный объект}) < Z_{\min}(\text{пробное тело})$, то протыкание невозможно. Перейти к следующему телу. В противном случае:

Проверить видимое протыкание.

Если $Z_{\max}(\text{пробный объект}) > Z_{\max}(\text{пробное тело})$, то пробный объект может проткнуть переднюю грань пробного тела.

Установить флаг видимого протыкания для последующего использования. Занести проткнутое тело в список протыканий.

Если $X_{\max}(\text{пробный объект}) > X_{\min}(\text{пробное тело})$ или $X_{\min}(\text{пробный объект}) < X_{\max}(\text{пробное тело})$, то пробный объект может проткнуть бок пробного тела.

Установить флаг видимого протыкания для последующего использования. Занести тело в список протыканий.

Если $Y_{\max}(\text{пробный объект}) > Y_{\min}(\text{пробное тело})$ или $Y_{\min}(\text{пробный объект}) < Y_{\max}(\text{пробное тело})$, то пробный объект может проткнуть верх или низ пробного тела.

Установить флаг видимого протыкания для последующего использования. Занести проткнутое тело в список протыканий.

Если список протыканий пуст, то устанавливать флаг протыкания не надо.

Провести проверки экранирования ребер:

Вычислить s и d для ребра.
Вычислить p , q , w для каждой плоскости, несущей грань пробного тела.

Проверка полной видимости. Если ребро полностью видимо, то перейти к следующему ребру.

Сформировать уравнения $h_j = 0$ и решить их, объединяя попарно и включив в систему уравнения границ $t = 0$ и $t = 1$. Если установлен флаг видимого протыкания, то в систему надо включить и уравнение границы $a = 0$. Запомнить точки протыкания. В противном случае границу $a = 0$ не учитывать.

Для каждой пары (t, a) , являющейся решением, проверить выполнение условий $0 \leq t \leq 1$, $a \geq 0$ и $h_j > 0$ для всех других плоскостей. Если эти условия выполнены, то найти T_{\max_min} и T_{\min_max} .

Вычислить видимые участки отрезков и сохранить их для последующей проверки экранирования телами с более низкими приоритетами.

Определить видимые отрезки, связывающие точки протыкания:

Если флаг видимого протыкания не установлен, перейти к процедуре визуализации.

Если точек протыкания не обнаружено, перейти к процедуре визуализации.

Сформировать все возможные ребра, соединяющие точки протыкания, для пар тел, связанных отношением протыкания.

Проверить экранирование всех соединяющих ребер обоими телами, связанными отношением протыкания.

Проверить экранирование оставшихся соединяющих ребер всеми прочими телами сцены. Запомнить видимые отрезки.

Визуализировать оставшиеся видимые отрезки ребер.

Заметим, что этот алгоритм можно реализовать и с обратным списком приоритетов. Оценки по времени продемонстрировали почти линейно зависящий от числа брусков рост, причем последнее число достигало 1152 .

[Обсудить на форуме »](#)

На правах рекламы

Качественные [напольные часы](#) [купить](#) за 1 час.