УДК 517.934

© **Д. К. Михалёв**

ИСПОЛЬЗОВАНИЕ 2^n-МЕРНЫХ КУБИЧЕСКИХ ДЕРЕВЬЕВ ДЛЯ РЕАЛИЗАЦИИ ПИКСЕЛЬНОГО МЕТОДА ВЫЧИСЛЕНИЯ МНОЖЕСТВ ДОСТИЖИМОСТИ В n-МЕРНОМ ПРОСТРАНСТВЕ И ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ[1]

Доклад посвящён организации хранения информации о множестве достижимости управляемых систем [1]. Такое представление информации можно использовать для эффективной реализации пиксельного метода вычисления множеств достижимости.

§ 1. Описание способа хранения

Рассмотрим куб со стороной длины 2^m, где m — целое число, определяющее используемое количество младших бит каждой координаты; поделим каждое ребро пополам. Это деление определит 2^n более мелких кубов. Для этих кубов осуществим аналогичное дробление вплоть до кубов со стороной 1. Таким образом, мы получим дерево вложенных n-мерных кубов. Для двухмерного случая такое дерево называется *quadtree*, а для трёхмерного — *octree* [2,3].

При переходе от текущего узла-куба со стороной 2^α к потомку — подкубику со стороной $2^{\alpha-1}$, используются биты с номером $\alpha-1$ каждой координаты, и происходит уточнение местоположения в пространстве. После перехода α уменьшается на единицу. Предполагается, что в структуре дерева вместо ссылок на узлы могут встречаться пустые ссылки (подкуб — пуст), а также отметки о целиком заполненных подкубах, тогда вместо ссылки на куб хранится сама информация о кубе (подкуб заполнен однородной информацией).

§ 2. Занесение объекта в дерево

Для добавления в такое дерево заполненного куба со стороной длины 1 (в двухмерном случае пикселя) необходимо:

1. Перейти к ближайшему кубику, существующему в дереве, у которого значащие биты координат совпадают с искомыми координатами; поиск можно вести двигаясь с вершины (уточнением) или поднимаясь от любого кубика с известными координатами. В некоторых случаях требуется создавать новую вершину (с большим m) — куб, который пространственно содержит все остальные кубы дерева.

2. Спуститься (переходить к кубам-потомкам) в соответствии с искомыми координатами до куба со стороной 2, или пока не встретим заполненный куб. После спуска записываем (в соответствии с последним битами каждой координаты) в поле ссылки на куб-потомок значение заполненного куба.

3. Анализируем все указатели на потомков в текущем кубике. Если все они хранят одинаковое значение заполненного куба, то куб удаляется и в его предке вместо ссылки на него ставится то же самое значение заполненного куба. Поднимаемся на один уровень и повторяем эту процедуру, и так далее вплоть до вершины. Этим самым мы освобождаем память, занятую ненужной однородной информацией.

Повторяя эту процедуру многократно, можно занести весь объект, представленный множеством кубиков со стороной длины единица (в двумерном случае — пикселей).Такой способ позволяет минимизировать память, используемую для хранения заполненных объектов, сохраняя достаточно быстрый (логарифмический) доступ к каждому кусочку объекта.

§ 3. Определение границы множества

Это представление позволяет быстро достроить и выделить граничные (не имеющие соседа при изменении любой координаты на 1) кубики. Это осуществляется в 2 прохода:

[1]Работа выполнена при поддержке РФФИ (гранты 05-01-00601, 04-01-96099-р2004урал_а), гранта Президента РФ по поддержке ведущих научных школ НШ-8512.2006.1 и программы научного сотрудничества с СО РАН.

1. Для каждого кубика-узла, начиная с листьев и кончая вершиной дерева, определяется и сохраняется факт наличия пустот на каждой грани (по 2 грани на каждую координату).

2. Для каждого полностью заполненного кубика (вместо ссылки на блок памяти стоит значение (например цвет)), начиная с вершины дерева, проверяются 2 соседа (справа и слева) по каждой координате того же уровня (яруса) α, если соответствующая грань соседа содержит пустоту, тогда этот кубик считается граничным, и если $\alpha \neq 0$ (сторона куба больше 1), тогда узел создаётся и структура дерева соответствующим образом обновляется. В созданном узле вместо всех ссылок на подузлы ставится то же значение заполненного кубика и отмечается, что на каждой грани пустот нет.

Возможность выделять границу позволяет:

1. Визуализировать только границу.

2. В пиксельном методе (точнее его модификации для n-мерного случая) делать расчёт множества достижимости только от границы множества на предыдущем шаге, что при некоторой модификации алгоритма может дать существенный рост скорости при том же результате.

§ 4. Визуализация множества

Представление множества в виде такого дерева позволяет также эффективно визуализировать трёхмерное множество (без рассмотрения всех кубов со стороной 1).

Визуализация осуществляется проходом по дереву от вершины к листьям, при этом:

1. Каждый рассматриваемый узел-кубик оценивается на попадание в пирамиду видимости и, если он не виден, то его подкубики-потомки не рассматриваются.

2. Оценивается видимый размер (радиус в пикселях на экране) каждого рассматриваемого кубика и, если размер больше допустимого, то рекурсия по дереву продолжается, а если меньше, то выводится круг соответствующего размера, цвет которого является средневзвешенным из всех подкубиков данного кубика (он вычисляется предварительно).

3. Структура дерева позволяет производить частичную сортировку при проходе по дереву от вершины к листьям, такую, что кубики, попадающие в одни и те же точки экрана, анализируются в строго определённом порядке (например, с возрастанием расстояния от камеры). Эта сортировка выполняется локально (с учётом центров подкубиков и направления сортировки) для непосредственных потомков каждого кубика и может выполняться с помощью предварительно посчитанной таблицы. Также возможно использование иерархического Z-буфера, это позволяет отсекать кубы, которые имеют большой видимый размер, но целиком закрыты другими частями объекта.

4. Возможно использование аппаратных средств вывода графики современных видеокарт. В этом случае можно кешировать в памяти видеокарты области пространства, относящиеся к одному кубу определённого размера и его видимому размеру (мип-уровню). Это позволяет существенно (в 5 раз) увеличить производительность визуализации.

Список литературы

1. Гусейнов Х. Г., Моисеев А. Н., Ушаков В. Н. Об аппроксимации областей достижимости управляемых систем // Прикл. мат. и мех., 1998, Т. 62., вып. 2. С. 179–187.
2. Wilhelms J., Gelder A.V. Octrees for faster isosurface generation. // ACM Transactions on Graphics. July 1992. V.11 № 3. P. 201–228.
3. Kyu-Young Whang, Ju-Won Song, Ji-Woong Chang, Ji-Yun Kim, Wan-Sup Cho, Chong-Mok Park Il-Yeol Song Octtree-R: An adaptive octree for efficient Ray Tracing // IEEE Transactions on Visualization and Computer Graphics. 12. 1995 V.1. №4 P. 343–349.

Михалёв Дмитрий Константинович
Уралький государственный технический ун-т,
Россия, Екатеринбург
e-mail: ddmk@r66.ru