УДК 681.3:771.537.442

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ОРГАНИЗАЦИИ ДИНАМИЧЕСКОГО ФОРМИРОВАНИЯ МАССИВА ДАННЫХ В ПРОЦЕССЕ РАБОТЫ АЛГОРИТМА ИТЕРАЦИОННОГО УТОЧНЕНИЯ ТРЁХМЕРНОЙ ОБЛАСТИ

Толок Н.Б., к.т.н., доцент, Толок А.В., к.т.н., доцент

Запорожский государственный университет

ВВЕДЕНИЕ

Проблема организации динамических массивов для хранения и обработки информации, итерационно изменяющейся в процессе своего формирования, являлась и является одной из интереснейших задач моделирования данных [1,2]. Примером подобной проблемы служит задача построения динамически формируемого массива данных при рекурсивном уточнении рассматриваемой области. Итерационные подходы уже используются в визуальном анализе когнитивной компьютерной графики при моделировании сложных объектов, представленных аналитическим или кусочно-аналитическим методом (R-функции) [3,4]. Основным недостатком подобных построений является низкая производительность алгоритмов по причине многократной повторяемости действий над каждой точкой исследуемой области объекта. Достоинством служит пространственная организация уточняемых данных для каждой итерации, что позволяет получить в результате не только изображение объекта при заданных характеристиках, но и массив данных, содержащий матричное представление «поведения» исследуемой «сцены объекта». В отличие от традиционных способов отображения сцен, которые при любых изменениях отношения наблюдателя к объекту требуют перепостроения изображения [5], полученный в результате итерационного подхода массив позволяет разработать алгоритмы динамического отображения исследуемого объекта в различных ракурсах и сечениях. При этом достаточно базироваться лишь на алгоритме пересортировки массива и пересчёта матрицы поворота в пространстве сцены.

Интерес представляет разработка удобной внутренней и логической организации такого массива, позволяющей просто управлять итерационными данными, особенно на этапах рекурсивного перехода.

1. ВНУТРЕННЯЯ ОРГАНИЗАЦИЯ РЕЗУЛЬТАТОВ РАСЧЁТА АЛГОРИТМА ИТЕРАЦИОННОГО УТОЧНЕНИЯ ПОСРЕДСТВОМ ПОЛОВИННОГО ДЕЛЕНИЯ

Количество подобластей для половинного деления области равно 2^n, где n - размерность области разбиения. Таким образом, отрезок подразделяется на 2^1 элемента, квадрат на 2^2, куб уже можно представить 2^3, т.е. восьми элементным разбиением и т.д.. Рассмотрим случай обработки разбиением трёхмерной сцены E^3, принципы которого легко переносимы на более низкие, а также на более высокие размерности исследуемых объектов.

Количество элементов трёхмерного массива l, содержащего результаты расчёта алгоритма итерационного уточнения посредством половинного деления, зависит от количества итераций m и рассчитывается как $l = 2^{nm}$. Определим индексную организацию формируемого массива в трёх направлениях (i, j, k). Длина индексного направления, выбранного вдоль одной из осей координат исследуемой сцены, составляет $\sqrt[3]{2^{nm}}$ единиц. Таким образом, размер массива определяется как $i \times j \times k$, где $i = j = k = [0,...,\sqrt[3]{2^{nm}} - 1]$. Увеличение размера массива обратно пропорционально размеру конечного элемента области исследования. В ходе организации таких данных особый интерес представляет промежуточный этап – переход к следующей итерации уточнения, который должен содержать в себе как данные предшествующего этапа исследования области, так и данные исследуемой области на новом этапе уточняющего разбиения. При этом возможно применение известных методов организации динамических структур иерархического дерева, позволяющих хранить информацию каждого этапа на своём уровне иерархии [1,2]. Недостатком таких структур является сложность

восприятия целостности рассматриваемого объекта, а также избыточность информации, необходимая лишь для общей увязки структуры «сверху вниз». В действительности, в таких случаях нас интересует только текущая информация для перехода от одной итерации к другой, при этом все предшествующие этапы разбиения исключаются из рассмотрения.

В работе предлагается описание динамического массива данных размерностью 3×3, позволяющего поддерживать целостность восприятия исследуемой трехмерной сцены с компактным заполнением данных, учитывая возможность хранения сразу обоих уровней разбиения в процессе перехода от одного уровня итерации к другому.

2. ДИНАМИЧЕСКИЙ МАССИВ ДЛЯ ХРАНЕНИЯ РЕКУРСИВНО УТОЧНЯЕМЫХ ТРЁХМЕРНЫХ СЦЕН

По своей сути предложенное представление обыкновенного трёхмерного массива отличается своей физической и логической организацией из-за специфичности поставленной задачи. Заполнение массива осуществляется в обратном порядке, по сравнению с традиционным, и начинается с элемента, содержащего индексы $i = j = k = \sqrt[3]{2^{nm}}$. Причины, по которым был выбран именно такой порядок, мы обсудим позднее. Предварительно рассмотрим начальные этапы формирования динамического массива.

На первом этапе трехмерный элемент (на рисунке - куб) подразделяется на восемь равных элементов (рис. 1). Таким образом, для организации индекса нам потребуется 3 бита (от 000 до 111). Проиндексируем массив так, чтобы последовательность нулей и единиц отражала трехмерное положение элементов объекта в соответствии с направлениями осей i, j, k. На рисунке видно, что нулевое значение соответствующего бита в значениях индексов $[ijk]$ определяет расположение куба непосредственно на координатной оси. Логический номер поэлементного заполнения m для рассмотренного случая можно определить как

$$m = 2^n - o_{10}(i, j, k),$$

где $o_{10}(i, j, k)$ - функция перехода к десятичному значению комбинации двоичного кода, образуемого группировкой значений элементов $[ijk]$.

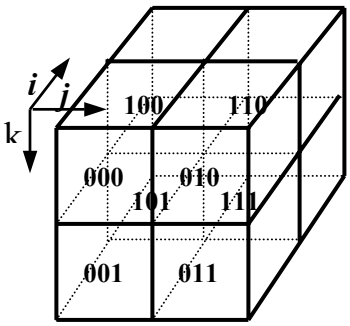


Рис.1. Пример первого уровня индексного разбиения динамического массива

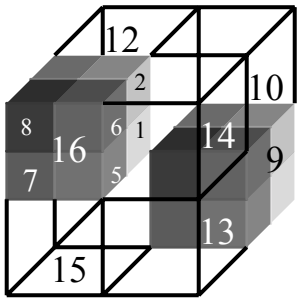


Рис.2. Организация логической последовательности заполнения динамического массива «от последнего элемента к первому».

На втором этапе уточнения трёхмерной сцены осуществляется разбиение областей, которым в массиве соответствуют элементы $000...111$, на восемь равных элементов методом половинного деления. Рис. 2 отображает динамическое развитие матрицы при переходе ко второму этапу уточнения трёхмерной области. При этом происходит увеличение размерности динамического массива в два раза вдоль каждой оси. Для наглядности на рис. 2 логическая очередь заполнения массива отражена последовательным рядом значений, где с 9 по 16 - укрупнённые делимые элементы на последующей итерации рассматриваемой области определения. Элемент 9 является «итерационным потомком» элемента 1, поскольку отражает ту же самую область, только уже в уточненном виде за счёт деления «итерационного предка» на восемь частей. Процесс итерационного уточнения трёхмерной области предусматривает последовательную обработку «итерационных предков» 1-8 с последовательным отражением результатов деления в «итерационных потомках» 9-16.

Такая организация заполнения позволяет сохранять данные, полученные на предыдущей итерации в первых ячейках матрицы до тех пор, пока не заполнятся все свободные ячейки значениями следующей итерации. Далее, элементы текущей итерации (16) «вытесняют» значения предыдущей итерации (1-8) в первых ячейках куба, и происходит переход к новой итерации с соответствующим увеличением массива. Это позволяет сохранять связь с предыдущей итерацией до требуемого предела.

Здесь возникает проблема структурной организации индексных данных динамического массива, позволяющей производить увязку между индексным распределением элементов массива и их логическим порядком. Для этого снова обратимся к бинарному коду.

3. БИНАРНАЯ СТРУКТУРА ПРЕДСТАВЛЕНИЯ ДИНАМИЧЕСКОГО МАССИВА ДАННЫХ

В предыдущем разделе мы совершенно удачно заметили, что индексная организация восьмиэлементной ячейки может быть очень удобно представлена комбинациями двоичного трёхбитового кода. Идея пространственной компоновки данных в рекурсивно возрастающем массиве 3x3, отображенная на рис. 2, позволяет предложить описание развития итерационной структуры в том же бинарном виде. Структурность бинарного представления чисел позволяет создать промежуточное представление данных логического заполения в виде двоичного кода.

Значения от **0** до **7** посредством трёхбитовой комбинации описывают разбиение элемента на подэлементы. Значения от **8** до **127** организуют шестизначный код, где первая тройка значений организует элементы глобального куба, а вторая тройка значений – элементы локальных кубиков, полученных в результате деления элемента глобального куба (рис.3). Таким образом, структура битового представления динамического массива может быть описана в виде последовательности трёхбитовых комбинаций, где количество таких комбинаций определяет уровень вложенности рекурсии в ходе динамического заполнения итерационно возрастающего массива. Например, битовая комбинация ***001 011 001*** означает второй куб на третьем уровне уточнения, четвёртый куб на втором уровне уточнения и второй куб на первом уровне уточнения. На рис. 4 результирующий элемент отмечен самым тёмным тоном.

000	001	...	...	111
000	000	...	...	000
001	001	...	...	001
010	010	...	...	010
011	011	...	...	011
100	100	...	...	100
101	101	...	...	101
110	110	...	...	110
111	111	...	...	111

Рис.3. Пример битового описания структуры динамического массива для трёх итераций разбиения.

Очевидно, что элементы трёхмерного массива во внутреннем представлении компьютера традиционно представляются индексной адресацией (i,j,k). Например, значение, заносимое в элемент массива на рис. 4, определяется индексами массива $(0,2,7)$.

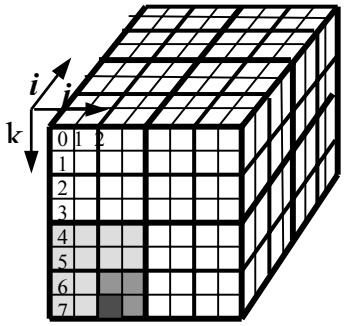


Рис.4. Пример положения элемента при битовом сочетании ***001 011 001***

Рассмотрим переход от предлагаемого бинарного представления к индексному представлению и наоборот. Индексное представление элемента массива определяется как:

$$i = \sum_{n=L/3}^{1} (2I_{3n})^{n-1},$$

$$j = \sum_{n=L/3}^{1} (2J_{3n-1})^{n-1},$$

$$k = \sum_{n=L/3}^{1} (2K_{3n-2})^{n-1},$$

где $I_{3n}, J_{3n-1}, K_{3n-2}$ - значения элементов для n-ой трёхбитовой комбинации,

$n = L/3$ - количество итераций алгоритма уточнения области, L - размерность вектора бинарного представления элемента динамического массива.

Например, для вектора элемента ***001 011 001*** индексное представление можно получить в виде:

$$i = (2\cdot 0)^2 + (2\cdot 0)^1 + (2\cdot 0)^0 = 0 + 0 + 0 = 0,$$

$$j = (2\cdot 0)^2 + (2\cdot 1)^1 + (2\cdot 0)^0 = 0 + 2 + 0 = 2,$$

$$k = (2\cdot 1)^2 + (2\cdot 1)^1 + (2\cdot 1)^0 = 4 + 2 + 1 = 7.$$

Для решения обратной задачи достаточно последовательно перевести десятичное представление индексов i, j, k в двоичный код и произвести группировку по столбцам в последовательные тройки по каждому элементу, как показано ниже:

0	=	0	0	0
2	=	0	1	0
7	=	1	1	1

В результате мы получим искомые трёхбитовые комбинации, отображающие положение элемента на всех итерационных уровнях.

ЗАКЛЮЧЕНИЕ

Предлагаемая организация данных относительно проста в управлении, характеризуется целостностью представления и компактна при итерационных переходах к удвоеной детализации. Данная модель позволяет собирать и хранить данные, формируемые в ходе итерационного уточнения исследуемой области и предлагается в использовании алгоритмов, связанных с рекурсивной динамикой.

ЛИТЕРАТУРА

1. Мейер Б., Бодуэн К. Методы программирования: В 2 т.: Пер. с фр. – М.: Мир, 1982. - Т.1. -356 с.
2. Кнут, Дональд, Эрвин. Искусство программирования: В 3 т.: Пер. с англ.- Т. 3: Сортировка и поиск. – М.: Издательский дом “Вильямс”, 2000. – 832 с.
3. Толок А.В., Мухин В.В. Алгоритм итерационного уточнения области исследования поверхности // Вісник Запорізького державного університету: Фізико-математичні науки. Біологічні науки. –1998. - N2. - С.90-97.
4. Гладкий Б.М., Толок А.В. Применение алгоритма частичной сортировки по глубине к визуализации функции трёх переменных // Вісник Запорізького державного університету: Фізико математичні науки. Біологічні науки. –1999.- N2.- 28-35С.
5. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ. – М.: Мир, 1989. – 512 с.