

УДК 550.8.05

ВЫБОР АЛГОРИТМА ПОСТРОЕНИЯ 3-D МОДЕЛИ ДЛЯ КРИВОРОЖСКОЙ СГ-8

Златокрылец Я. С., Агаркова Н. Г.
(УкрНИМИ НАНУ, г. Донецк, Украина)

Проаналізовано алгоритми побудування тривимірних моделей в прикладі до ГІС. Проведена оцінка алгоритмів за складністю та швидкістю побудови сцени. Обраний алгоритм октарних дерев для програмної реалізації трьохвимірної моделі Криворізької НГ-8.

Algorithms of building 3-D models in GIS were analysed. An estimation of algorithms by complication and speed of construction scene were made. An algorithm of oktar trees were chosen for program realization for 3-D model of Kryvorojkaya SG-8.

При широком применении компьютеров в современной геологии уже теоретически возможно создание объемных геологических карт для любой геологической среды в виде трехмерных моделей. Такого рода работы ведутся во многих странах Западной Европы и Америки, но пока они имеют коммерческий характер и не используются геологическими службами этих стран, а используются нефтяными и рудными компаниями. В настоящее время в мировой программной индустрии уже разработано несколько систем, обеспечивающих трехмерное представление геологических структур. Практически все они ориентированы на решение задач нефтегазовой геологии. В качестве примера можно назвать линейку программных продуктов моделирования резервуаров (GeoSurf, GeoSim, Heresim, SimGrid, SimUp) французской компании Beicip Franlab, GMP (Stratamodel Inc., Texas), Petrel (Technoguide, Norway)). Такие программы могут совмещать

в трехмерном пространстве данные сейсмических профилей и скважин, а также строить поверхности слоев на основании этих данных. Геологическая модель в таких программах существенно трехмерна и обладает всеми преимуществами такого представления.

Алгоритмы подобных программ ориентированы на очень хорошую изученность территории геологическими и геофизическими методами – предполагается, что для изучаемой территории существует плотная сеть скважин глубокого бурения и сейсмических профилей или трехмерная сейсмика. [9].

Среди немногочисленных русскоязычных источников, где уделяется внимание базовым концепциям геометрического моделирования в геологии, следует отметить работы А.Н. Петрова и сотрудников возглавляемой им лаборатории, работающих при ДВО РАН. Они занимаются различными аспектами тектоники и структурной геологии, используя математические методы моделирования [1]. Эти разработки могут быть использованы в качестве базового математического аппарата при программной реализации трехмерных геологических моделей.

В работе [2] изложен материал по широчайшему спектру вопросов структурной геологии. Причем, все данные для различных методик сводятся к единой трехмерной координатной форме, что делает возможным комбинирование различных приемов геометрических построений и создание эквивалентных им аналитических выражений.

Что касается геометрического моделирования с применением ПК в структурной геологии, здесь следует сказать, что в целом данное направление развито слабо. Как правило, большинство работ представляют собой описание того, как производились те или иные структурные построения для какого-либо конкретного геологического объекта. Не наблюдается какого-либо единого подхода к построению, единой отправной базы понятий.

В 2005-2008 гг. силами УкрНИМИ была создана электронная открытая реляционная иерархическая база данных Криворожской сверхглубокой скважины СГ-8 (глубина 5432 м) в которой собрана уникальная информация, полученная в ходе бурения скважины. Информационные блоки базы данных содержат ре-

результаты геологических, минералогических, геохимических, лабораторных, геофизических и др. видов исследований. База данных СГ-8 - это не просто набор данных, а автоматизированная система управления, позволяющая получать быстрый доступ к данным и их статистическую обработку в заданном глубинном, возрастном интервале или по типам пород. База, в силу иерархической реализации, легко расширяема и может быть легко дополнена новыми данными без ущерба для целостности. Фрагмент геологической колонки приведен на рисунке 1.

Бурение и комплексные исследования СГ-8 сопровождались работами по геологическому изучению района ее заложения. В ходе проведения работ по геологическому изучению площадей (ГИП-1, 2, 3) был накоплен значительный объем геолого-геофизической информации, которая является основой при изучении разреза палеопротерозойской криворожской серии Криворожско-Кременчугской шовной зоны, перспективной на выявление рудопроявлений благородных и цветных металлов.

Geol_kolonka_new : таблица									
	Nom_int	Kod_stv	Glub_ot	Glub_do	m_os	Kern	Strat_ind	Nom_obr	Nom_prob
▶	100	4	0	56,3	56,3	37,5	52		
	200	4	56,3	161,2	104,9	71,4	53		
	300	4	161,2	171	9,8	5,6	8 32-33	271	
	400	4	171	183	12	9,5	8 62-166	52337, 52330	
	500	4	183	194	11	11	8 167-230	52339	
	600	4	194	210	16	12,3	8 231-366	52340-52344	
	700	4	210	224	14	11,3	9 367-455	52344-52348, 4	
	800	4	224	232	8	6,7	9 456-482	52349	
	900	4	232	270	38	31,4	9 483-699	52350-52352, 4	
	1000	4	270	278,3	8,3	7,8	9 700-711	52353	
	1100	4	278,3	327	38,7	31,4	9 712-983	52354-52358, 8	
	1200	4	327	405	78	67,3	9 984-1351		
	1201	4	327	341	14		9 984-1068	52359	
	1202	4	341	345	4		9 1069-1090	52360	
	1203	4	345	347	2		9 1091-1103		
	1204	4	347	383	36		9 1104-1275	52361-52365	
	1205	4	383	405	22		9 1276-1361	52366, 52367	

Рис. 1. Фрагмент БД с исходными данными по скважине СГ-8 (привязки по глубине, стволу, пробам, образцу, керну)

Расширение базы данных СГ-8 и пополнение её результатами работ, проведенных при выполнении ГИП-1, 2, 3, предоставит

быстрый доступ к данным по скважинам для проведения анализа и геологической интерпретации, а так же позволит формировать из скважин как реальные профили, так и виртуальные (рис. 2). Разрезы могут быть как геологические, так и распределённые по различным параметрам (геохимическим, геофизическим и т.д.), построенные в изолиниях. Таким образом, электронная геологическая карта может быть дополнена разрезами: одномерными по скважине или двумерными по профилям.

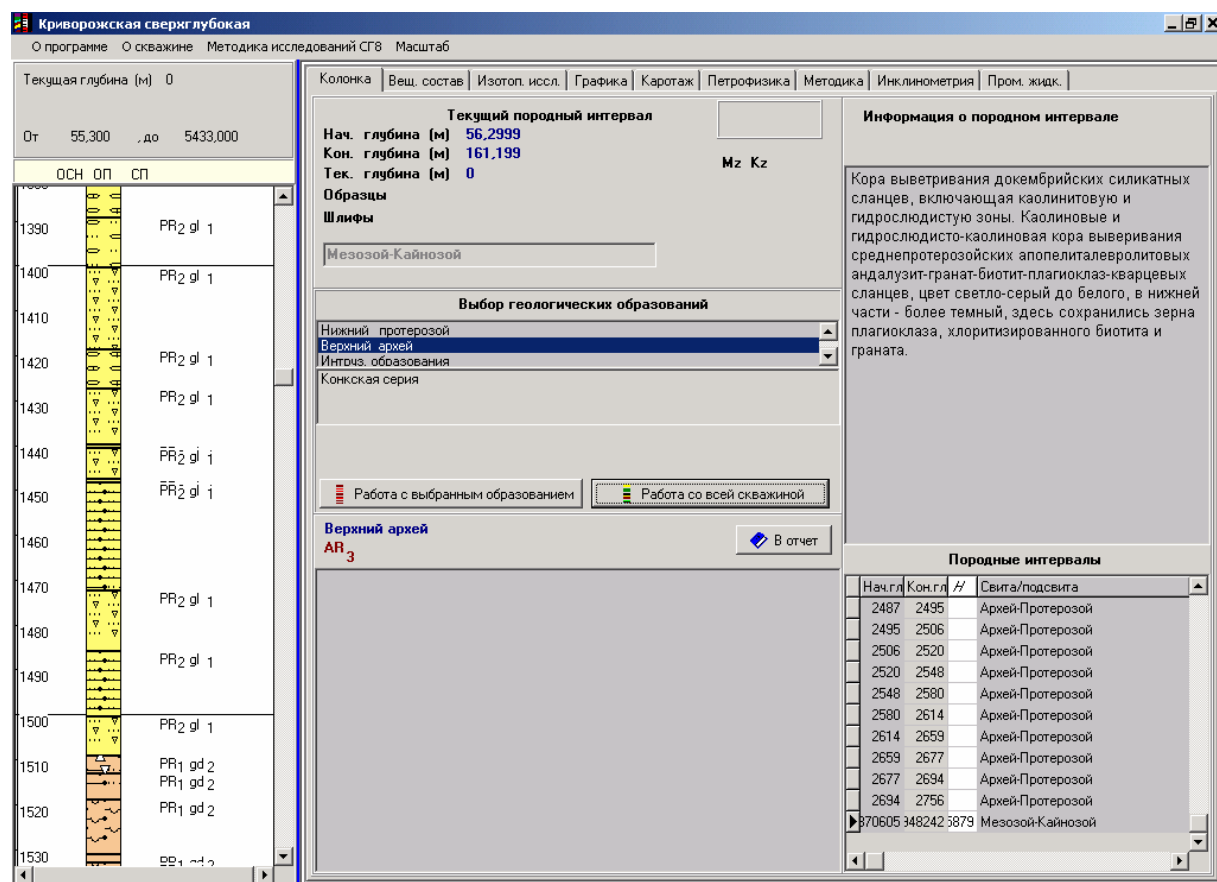


Рис. 2. Фрагмент программы с данными по СГ-8

Однако часто одномерных скважин и двумерных профилей оказывается недостаточно. Эффективность анализа и обработки данных существенно возрастет, если будет разработана их цифровая трехмерная модель, опирающаяся на базу данных. Информация по скважинам содержит числовые и текстовые данные по: координатам устьев скважин, инклинометрии, литологии, стратиграфии, каротажу, лабораторным и аналитическим исследовани-

ям с глубиной привязкой. Следовательно, модель по скважинам может быть реализована в 3-х мерном пространстве, т.к. данные имеют координатную привязку (x, y, z), рассчитанную согласно инклинометрии скважины (рис. 1).

Алгоритмы и методы пространственного моделирования

Трёхмерное моделирование вплотную связано с представлением пространственных отношений, что является фундаментом большой группы операций, которые должны выполняться в ГИС: включение, смежность, равенство, направление, пересечение, связанность и т.д. Были рассмотрены и проанализированы некоторые из методов представления 3D моделей:

1. Симплициальный комплекс.
2. Сеточная модель.
3. Каркасная модель
 - сплошная или замкнутая;
 - цифровая поверхностная;
 - сложная поверхностная.
4. Построение поверхностей по триангуляционным нерегулярным сетям (TIN)
5. Векторизация данных методом создания формальной структура 3D векторных данных.
6. Алгоритм разбиения пространственных данных на иерархические древовидные структуры - октарное дерево.

Октарное дерево - это универсальное решение, которое позволяет увеличить скорость перерисовки почти всех типов и видов данных ГИС, начиная с рельефов, зданий, отдельных объектов, деревьев, данных для геологических исследований и т.д.

Суть алгоритма построения октарного дерева заключается в том, при построении такого дерева вся сцена вписывается в куб, и этот куб делится тремя плоскостями на 8 одинаковых кубов (рис. 3). Каждый из получившихся кубов делится аналогично пока не станет содержать менее определённого количества графических примитивов. Все примитивы, ограничиваемые этим кубом, присваиваются ему. Каждый куб считается узлом дерева. Такой узел, который содержит только примитивы, но не содержит других узлов, называется листом. Графические примитивы –

это, как правило, треугольники, на которые можно разбить все фигуры в сцене.

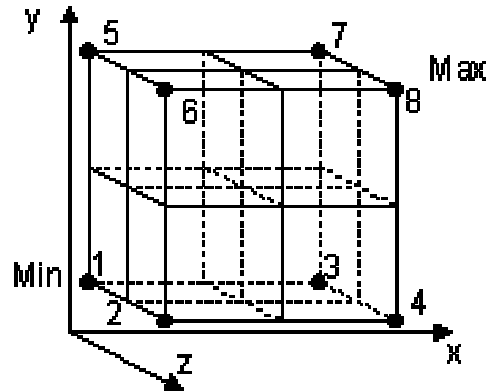


Рис. 3. Точки, на основании которых принимается решение о видимости узла

В итоге, получается иерархическая структура представления данных, где корневой узел - куб, в который вписана вся сцена, а листьям присваиваются все примитивы сцены. При визуализации таких структурированных пространственных данных, делается тест, видим ли корневой узел (куб), если нет, то и вся сцена не видна, и выводить на экран ничего не нужно. Если он видим, то делается тест видимости его дочерних узлов. И так до тех пор, пока не будет сделан тест видимости листов. Если лист видим, то принадлежащие ему примитивы выводятся на экран. Если какой-то узел не видим, то тестирование всех его потомков на видимость не производится, и принадлежащие им примитивы на экран не выводятся.

Таким образом, рассматриваемые алгоритмы позволяют увеличить скорость вывода сцены путём отсечения групп примитивов, не входящих в поле зрения наблюдателя. Они могут работать со всеми наборами пространственных данных, встречающимися в ГИС, и это большое преимущество, по сравнению с другими алгоритмами, увеличивающими скорость вывода на экран - например, алгоритмы плавающего горизонта, теневой и др.

Кроме октарных деревьев, применяются бинарные деревья - их особенность построения в том, что сцена сразу вписывается в параллелепипед, его разбиение происходит одной плоскостью, на

два одинаковых параллелепипеда, пока каждый не содержит менее определённого количества примитивов.

Суть всех весовых деревьев в том, что секущие плоскости проходят через весовую точку. Весовая точка находится как среднее арифметическое точек всех примитивов, входящих в узел.

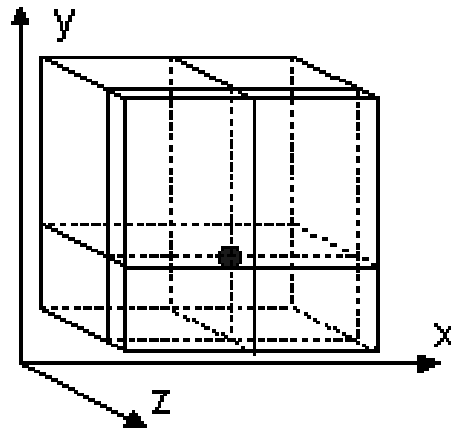


Рис. 4. Весовая точка

Например, на рис. 4, выделена точка (весовая), через которую проходят три секущие плоскости. В результате получают восемь параллелепипедов. Продолжив разбиение сцены получим алгоритм октарного весового дерева.

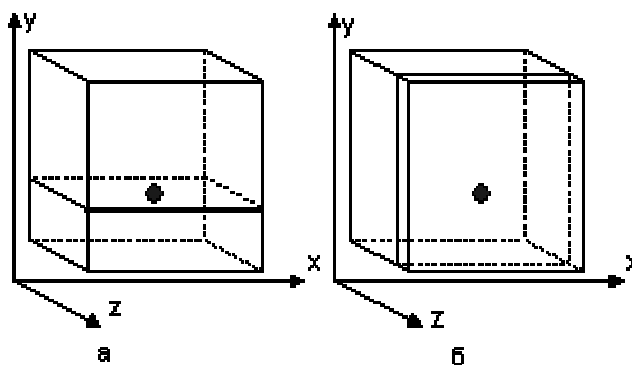


Рис. 5. Пример разбиения сцены на бинарное весовое дерево

Если сцену вписать в параллелепипед, и через его весовую точку провести одну секущую плоскость, получим два параллелепипеда. Каждый полученный также делим по весовой точке

ещё на два, и так до тех пор, пока не выполнится условие прекращения деления (рис. 5). Это алгоритм весового бинарного дерева.

В табл. 1 представлены обобщённые достоинства и недостатки применения бинарных весовых деревьев при выводе на экран трёхмерной сцены.

Таблица 1
Обобщённые достоинства и недостатки применения деревьев при выводе на экран трёхмерной сцены

Достоинства	Недостатки
Увеличение скорости вывода на экран. Уменьшение объёма информации, поступающей на конвейер графического ускорителя, что даёт возможность эффективно использовать технологию разбиения пространства на деревья, в компьютерных сетях, системах с удалённым сервером. Многие алгоритмы увеличения скорости вывода на экран тесно связаны с бинарными деревьями.	Увеличение размера памяти, для хранения структуры дерева трёхмерной сцены. Увеличение времени запроса ГИС данных сцены из памяти. Многие модификации построения деревьев предполагают использование статической системы данных. Как правило, необходимо заранее формировать структуру дерева, перед выводом на экран. Сложный алгоритм, и как следствие - сложность применения некоторых видов анализа и алгоритмов увеличения скорости вывода на экран.

Особенности программной реализации.

Алгоритм разбиения пространства имеет в реализации ряд спорных и сложных моментов, которые разработчики решают на своё усмотрение.

Например, предлагается рациональное решение - сделать возможность присвоения примитивов не только листьям, но и узлам деревьев. Тогда, если примитив не может быть однозначно присвоен листу, он должен быть присвоен узлу, которому при-

надлежит этот лист, но при этом примитив должен однозначно вписываться в границы этого узла.

Еще одна проблема традиционного алгоритма - примитивы нельзя относить к узлу только по координатам их вершин. Эта проблема не существует в модифицированном алгоритме, так как там примитивы однозначно присваиваются узлам.

Создание слоёв в древовидной структуре.

Кроме того, в такой иерархической структуре ГИС данных необходимо предусмотреть наличие слоёв. Возникает проблема - как правильно организовать визуализацию слоёв, используя дерево.

Предлагаются следующие варианты:

1. Создать несколько деревьев - по одному для каждого слоя.
2. Указывать атрибут слоя для каждого примитива, и потом проверять, видим ли слой, и если да, то выводить примитив на экран.
3. Указать атрибут слоя для группы примитивов, и потом проверять, видим ли слой, и если да, то выводить группу примитивов на экран.

Первый вариант достаточно сложен, если у нас много слоёв, то придётся создавать много деревьев, и каждое проверять, видимо ли оно. Второй вариант проще, в большинстве случаев должен работать быстрее. Третий вариант - дерево, у которого листья (или группа примитивов, что в данном случае почти одно и то же) имеют атрибут слоя. Этот способ построения дерева оптимален. Он лучше первого способа тем, что позволяет избежать движения от корня деревьев к листьям раз, где - количество слоёв. И лучше второго способа тем, что проверка на необходимость вывода на экран невидимого слоя производится над группой примитивов целиком, а не по отдельным примитивам.

Модификации бинарных деревьев.

Для разделения сцены на бинарное дерево по весовому принципу, нужно найти весовую точку, и провести по ней секущую плоскость, как показано на рис. 6. В этом случае получится два параллелепипеда, с примерно одинаковым числом примити-

вов. Секущая плоскость может быть расположена под любым углом к горизонту.

Но если эти плоскости будут располагаться под любым углом, то в результате разбиения сцены будет разбита не на параллелепипеды, а на сложные трехмерные структуры. Получившиеся таким образом многоугольники будет сложно тестировать на попадание в пирамиду видимости, так как вместо анализа четырех вершин, придется производить анализ множества вершин на вхождение в пирамиду видимости. Кроме этого, увеличится объём памяти, для хранения вершин узлов. И усложнится алгоритм получения новых узлов. Поэтому предлагается разбивать сцену на параллелепипеды.

Для разбиения сцены на параллелепипеды, необходимо проводить вертикальные, или горизонтальные секущие плоскости. Если ограничиться только одной плоскостью (например, вертикальной), то разбиение получится неэффективным, такой вид разбиений можно применять, только если визуализируется рельеф.

Весовое разбиение пространство должно быть эффективно для сложно расположенных примитивов. Бинарное дерево выполняет ту задачу, а проверка на видимость узла и все остальные процедуры при визуализации одинаковы. Но преимущество октарного дерева в том, что оно разбивает пространство на кубы. Такое разбиение должно показывать максимальную скорость тестирования нахождения узла в пирамиде видимости. Так как кубы лучше других параллелепипедов отсекают невидимые группы примитивов, не входящих в пирамиду видимости, при любом положении наблюдателя. Действительно, если будет много узких длинных параллелепипедов, то при попадании хотя бы части одного из них в пирамиду видимости, будет выводиться на экран (или тестироваться на видимость) и другая часть, что будет уменьшать скорость вывода сцены. А задача исследуемых нами алгоритмов быстро и эффективно отсечь невидимые группы примитивов.

Чтобы тест на попадание в пирамиду видимости проходил как можно эффективней, нами предлагается делить узлы по такому принципу - параллелепипед должен стремиться кубу. Поэтому, если длинная сторона параллелепипеда расположена горизон-

тально - то делим её вертикальной секущей плоскостью. Если вертикально - то горизонтальной. То есть - делим секущей плоскостью наибольшую сторону параллелепипеда. Такое дерево мы назвали бинарным весовым.

Также существует ещё принцип деления узлов - проводится условное деление тремя возможными плоскостями. После этого, алгоритм проверяет соотношение вновь получившихся сторон, там, где соотношение окажется наиболее близким к единице, и проводится конечная секущая плоскость. Такое дерево мы назвали бинарным весовым пропорциональным.

Критерий определения листа.

Определять, является узел листом или нет, можно не только по количеству входящих в него примитивов, а и по размеру сторон куба узла. Если стороны меньше заданного - то это лист, иначе не лист. Но в наборах ГИС примитивы могут фактически находиться в одном месте, тогда алгоритм разбиения примитивов в иерархическую структуру может выйти в бесконечный цикл (или разбить сцену на большое количество узлов). Поэтому стоит применять ограничение на длину ребра куба листа, чтобы куб, который принадлежит листу, был не меньше заданного. Так же необходимо проверять разбиваемый массив на наличие дублирующих примитивов, иначе алгоритм разбиения на дерево может зациклиться.

При использовании деревьев вопрос, при каком максимальном количестве примитивов в узле, стоит прекращать деление узлов, стоит очень остро. Так, например, в бинарном дереве существует несколько максимумов скорости выведения сцены в зависимости от глубины разбиения. Поэтому, алгоритм должен учитывать количество примитивов в узле, и длину ребра, а в некоторых случаях, и среднюю длину ребра примитива. Именно таким сложным алгоритмом можно добиться максимально высокой скорости вывода в бинарных весовых алгоритмах.

Реализация алгоритма 3-D моделирования.

Для программной реализации были выбраны алгоритмы октарных деревьев. При различном положении наблюдателя в область видимости попадают разные части сцены, при этом процент видимости примитивов и скорость вывода меняются, что позво-

ляет объективно сравнить алгоритмы. Были проанализированы результаты полученных экспериментов. В качестве набора данных использовались результаты геологических исследований по Криворожской сверхглубокой (СГ-8).

Анализ структуры пространственных данных.

Скорость работы алгоритма зависит от того, как распределены примитивы по узлам. Были проведены аналитические оценки распределения примитивов по узлам, по равномерности разбиения, по количеству листьев, по количеству примитивов, по присвоенным узлам, которые не являются листьями (родительские узлы).

Минимальное и максимальное количество примитивов в листе бинарного весового дерева при малых значениях глубины разбиения выше, чем у октарного и октарного весового. Поэтому бинарное весовое дерево разбивает пространственные данные равномерней октарных деревьев. Действительно, пусть лист содержит 150 примитивов одного, видимого, и 150 примитивов невидимого слоя. Программа сразу приступит к выводу 150 примитивов видимого слоя на экран, и после установит, что остальные 150 примитивов невидимы, не проверяя атрибут каждого примитива. При этом выполнится примерно в 150 раз меньше операций, чем при проверке на видимость слоя каждого примитива.

Бинарное весовое пропорциональное дерево лучше всех рассмотренных алгоритмов разбивает пространственные данные, так значение минимального количества примитивов в листе у него выше, чем у других алгоритмов, что хорошо заметно при малых значениях глубины разбиения. Максимальное значение примитивов в листе у бинарного весового пропорционального дерева чуть отстаёт от бинарного весового, но это не так важно, поэтому равномерность разбиения пространства лучше всего у бинарного весового пропорционального дерева. Например, пусть для октарных деревьев количество примитивов, присвоенных родительским узлам равно 50000, при максимально допустимом количестве примитивов в узле до 200. Это значит, что 50000 примитивов отрисовывалось бы как минимум два раза вместо одного. Всего примитивов допустим 100000, положим, что при этом скорость обработки равна 1. Тогда, при использовании алгоритма, который

не присваивает эти примитивы родительским узлам, мы бы отрисовывали 150000 примитивов. Скорость обработки при этом была бы: $100000/150000=0,66$. То есть, потери составляли бы примерно 30 %. Это достаточно большая величина, тем более, что для бинарного дерева разница в скорости обработки ещё больше.

У октарного и октарного весового дерева количество листьев примерно одинаково, у бинарного весового в несколько раз меньше, а у бинарного весового пропорционального ещё меньше. Это свидетельствует об эффективном разбиении примитивов по узлам у бинарного весового пропорционального дерева.

В то же время у бинарных деревьев отношение количества листьев к количеству узлов в несколько раз меньше, и глубина иерархии больше, чем у октарных деревьев. Но это некритично, так как количество листьев у бинарных деревьев гораздо меньше, чем у октарных.

Оценка качества разбиения сцены алгоритмами.

Алгоритм неоптимально разбивает пространство если наблюдается очень малое количество делений по горизонтали сцены. Неравномерность разбиения в весовом бинарном дереве получается при неравномерном распределении примитивов. Алгоритм должен корректнее делить пространство, как это делает алгоритм бинарного весового пропорционального дерева.

При разбиении сцены алгоритмом бинарного весового пропорционального дерева сцена разбивается более равномерно, и параллелепипеды похожи на кубы. Безусловно, при таком разбиении, бинарное весовое пропорциональное дерево должно иметь более высокую скорость вывода сцены, чем бинарное весовое дерево. При видимости всех примитивов, все алгоритмы должны показывать примерно одинаковую скорость и эффективность обработки сцены.

Вообще говоря, скорость обработки алгоритма зависит от того, насколько удачно входят в пирамиду видимости узлы или листья. Отношение количества видимых узлов ко всем узлам, и видимых листьев ко всем листьям примерно одинаково при 30% видимости сцены. У октарного дерева и бинарного весового пропорционального дерева этот показатель в среднем на 10-15% ниже. Между отношением количества видимых узлов ко всем уз-

лам, видимостью листьев ко всем листьям и скоростью вывода сцены есть зависимость. Чем ниже эти показатели, тем выше будет скорость вывода сцены.

Алгоритм октарного дерева отлично зарекомендовал себя для увеличения скорости вывода примитивов на экран. Этот алгоритм с успехом используется ГИС, в играх и других прикладных графических программах. В целом, он обеспечивает более высокую скорость вывода сцены, чем у весового октарного и бинарного деревьев. Алгоритм бинарного весового пропорционального дерева имеет эффективную скорость вывода сцены примерно равную скорости отрисовки по алгоритму октарного дерева.

Структура алгоритма весового октарного дерева сложнее, чем у октарного дерева. У алгоритма октарного дерева больше время обработки данных, а следовательно и время вывода сцены. Время разбиения сцены на весовое октарное дерево больше, чем время разбиения на традиционное октарное дерево.

Алгоритм бинарного весового дерева универсален. Он применим ко всем наборам данных. Если для увеличения скорости при визуализации сложных помещений, лабиринтов, используют октарное дерево, то для увеличения скорости при визуализации рельефа используют квадродеревья. Скорость разбиения сцены этим алгоритмом выше, чем у октарного и октарного весового. И, если правильно подобрать максимально допустимое количество примитивов в листе, алгоритм весового бинарного дерева отстаёт от алгоритма октарного дерева на исследуемом наборе данных на 5-30%.

Недостатками этого алгоритма являются: сложность программирования, и подбор максимально допустимого количества примитивов в листе. А достоинства - универсальность применения к различным наборам данных. Используя этот алгоритм, можно получить высокую скорость обработки и вывода сцены при минимальном времени разбиения. Этот алгоритм является компромиссом между не очень большой скоростью и малым временем разбиения. Так как другие рассмотренные алгоритмы не обеспечивают такого малого времени разбиения и высокой скорости отрисовки сцены.

Если же ставится цель получить максимально возможную скорость обработки, то необходимо использовать октарное, или бинарное весовое пропорциональное дерево. Они должны иметь примерно одинаковые показатели скорости. Бинарное весовое пропорциональное дерево гораздо быстрее разбивает сцену, но у него достаточно сложно найти максимально допустимое количество примитивов в листе, при котором скорость будет максимальна. Если не критично время разбиения сцены, то лучше использовать октарное дерево. Если важно время разбиения сцены на дерево, (и если данные сложно расположены в пространстве - есть большие скопления и пустоты) то нужно использовать бинарное весовое пропорциональное дерево.

Выводы:

В статье проанализированы методы построения трехмерных геологических моделей и выбран метод для построения 3-D модели на основе базы данных по СГ-8. Предложена схема построения трехмерной модели на основании алгоритма октарных деревьев, состоящая из набора этапов:

- интерпретация данных бурения и выделение ключевых информативных данных;
- корреляция скважин, построение двумерных моделей по профилям, проведенным вдоль наборов скважин;
- анализ наборов двумерных моделей;
- построение трехмерной модели геологических границ и разломов путем корреляции двумерных моделей;
- построение трехмерной однородной решетки и идентификация узлов с геологическими структурами;
- программная реализация алгоритма октарного дерева и получение трехмерной модели СГ-8, с разбиением по слоям, породам, эпохам.

Результаты работы могут быть использованы при разработке и построении трехмерных моделей для почти всех типов и видов данных ГИС: скважин, шахт, рельефов, зданий, отдельных объектов, деревьев, данных для геологических исследований и т.д.

СПИСОК ССЫЛОК

1. Петров, А. Н. Геометрические модели слоистых тел и картирование сложнодислоцированных толщ путем интерполяции элементов залегания. Метод. рекомендации. [Текст] / А.Н. Петров. - Магадан, 1988.
2. Петров, А. Н. Математическое моделирование тектонических движений при изучении геологических структур. [Текст] / А.Н. Петров. - СВКНИИ ДВО РАН, 1997.
3. Применение персональных ЭВМ в геологических исследованиях. [Текст] / Отв. ред. С. Г. Бялобжеский, А. Н. Петров. - Магадан: - СВКНИИ ДВО РАН, 2001.
4. Васильков, Ю. В. Василькова Н. Н. Компьютерные технологии вычислений в математическом моделировании. [Текст] / Ю. В. Васильков, Н. Н. Василькова. - М., 2002.
5. Вистелиус А. Б. Основы математической геологии (определение предмета, изложение аппарата). [Текст] / А. Б. Вистелиус - Л., 1980.
6. Дэвис Дж. С. Статистический анализ данных в геологии. В 2 кн. [Текст] / Дж. С. Дэвис. - М., 1990.
7. Заика-Новацкий, В. С. Структурный анализ и основы структурной геологии. [Текст] Учеб. пособие. / В. С. Заика-Новацкий, А. Н. Казаков. - К., 1989.
8. Павлова, А. А. Начертательная геометрия. [Текст] учебник для студентов ВУЗов. / А.А. Павлова. - М., 1999.
9. Трехмерное геологическое картирование [Электронный ресурс] / Геологический факультет МГУ - Режим доступа: http://geo.web.ru/sbmg/geolok/mpr_00/rep2000.htm/ - 10.03.2009 г. - Загл. с экрана.