МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ ГЕОИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания

Красноярск 2003

УДК 528.9(07) M21

М21 Алгоритмы и структуры данных геоинформационных систем: Методические указания для студентов специальности 071903 – «Геоинформационные системы» / Сост. И.В. Варфоломеев, И.Г. Ермакова, А.С. Савельев. Красноярск: КГТУ, 2003, 34 с.

Печатается по решению редакционно-издательского совета университета

© КГТУ, 2003

Печатается в авторской редакции

Общие сведения

Методические указания подготовлены в соответствии с рабочей программой по курсу «Проектирование геоинформационных систем» для студентов Красноярского Государственного технического университета специальности 071903 – «Геоинформационные системы».

В методических указаниях отражены вопросы моделирования данных в геоинформационных системах, рассматриваются некоторые алгоритмы вычислительной геометрии. Применение структур данных и алгоритмов показано на примере моделирования поверхностей (рельефа).

В этих методических указаниях авторы преследуют две цели. Вопервых, студенты должны не только уметь работать с современным программным обеспечением ГИС, но и понимать, как внутри системы выполняется та или иная операция. Во-вторых, проектирование ГИС вовсе не ограничено использованием существующего коммерческого программного обеспечения.

Методические указания помогут студентам при выполнении лабораторных работ, в которых требуется на некотором языке программирования создать приложения обработки географических данных, сравнимые по функциональности с коммерческими ГИС, например MapInfo.

1. Структуры пространственных данных ГИС

1.1. Хранение растровых данных

В геоинформационных системах широко распространена растровая модель данных. Растры применяются для хранения и обработки данных дистанционного зондирования, для представления цифровых моделей рельефа, при визуализации геоданных и т.д. Существует множество вариантов кодирования растровых структур. Некоторые из них более экономно расходуют память, другие позволяют получать более быстрые алгоритмы. Растровая модель соответствует двумерному ячеистому изображению, которое хранится в памяти компьютера в виде одномерной последовательности значений. Растровые изображения обычно разлагаются по строке сверху – слева. Далее будут описаны другие способы эффективного представления растров.

В некоторых форматах графических файлов используется сжатие изображения, основанное на замене длительных последовательностей повторяющихся значений парой *<значение, количество повторов>* (рис. 1-а). Географические данные обычно автокоррелированны. В растровой модели это означает, что соседние ячейки имеют большую вероятность быть одинаковыми, чем разобщенные. При обычном порядке сканирования в конце каждой строки происходит скачок на начало следующей строки. Предложим простое изменение порядка сканирования. Нечетные строки будем кодировать слева направо, а четные – в обратном направлении (рис. 1-б). Направление сканирования напоминает движение быка, вспахивающего поле. Отсюда название этого способа сканирования – Boustrophedon (греч. – бык, вспахивающий поле). Теперь при переходе к новой строке первая ячейка является смежной последней ячейке старой строки. Так в линейном разложении растра сохраняется автокорреляция и повышается эффективность кодирования.

a)



АААА АВВВ ААВВ АААВ 16 байт 5A 3B 2A 2B 3A 1B 12 байт



АААА ВВВА ААВВ ВААА 16 байт 4A 3B 3A 3B 3A 8 байт

Рис. 1. Порядки сканирования растров, их линейное разложение и сжатие: а) обычный порядок; б) Boustrophedon.

Порядок сканирования Мортона (назван по имени Гая Мортона, впервые использовавшего этот способ в Canada GIS) основан на иерархическом разбиении карты. В предыдущих способах сканирования учитывалась автокорреляция значений ячеек только по одному направлению (по строке). Географические объекты образуют на растровом изображении пятна. В порядке Мортона предпринимается попытка сканирования ячеек таким образом, чтобы охватить линией обхода эти двумерные пятна.

Для растра размером 2 х 2 применяется обычный порядок сканирования. На следующем уровне матрица размера 4 х 4 складывается из четырех матриц размера 2 х 2, расположенных в таком же порядке, как ячейки матрицы 2 х 2 (рис. 2). Аналогично формируется линия сканирования любой матрицы порядка 2ⁿ. Матрица формируется уровень за уровнем, повторяя один и тот же шаблон размера 2 х 2. При сканировании растра по Мортону линия сканирования представляет собой фрактал. Недостатки сканирования по Мортону очевидны. Во-первых, присутствуют скачки, например, от ячейки 7 к ячейке 8. Во-вторых, таким способом можно кодировать только растры размера, кратного двум.

2	3	_	10	
)	1		8	
			2	
				ſ

10	11	14	15
8	9	12	13
2	3	6	7
0	1	4	5

Aī	- A -	. A .	` A	->
A	, Β.	<u> </u>	B	
A	- A -	B	B	
 A	` A	Ă	B	

42	43	46	47	58	59	62	63
40	41	44	45	56	57	60	61
34	35	38	39	50	51	54	55
32	33	36	37	48	49	52	53
10	11	14	15	26	27	29	31
8	9	12	13	24	25	28	30
2	3	6	7	18	19	22	23
0	1	4	5	16	17	20	21

AAAAABBBABAABBAA 5A 3B 1A 1B 2A 2B 2A

Рис. 2. Порядок сканирования растра по Мортону.

Рассмотрим следующий способ сканирования, в котором отсутствуют скачки между ячейками. На рис. 3 ячейки растра сканируется по линии Пеано. Имеется базовый П-образный шаблон, который поворачивается от уровня к уровню так, чтобы обеспечить непрерывность линии сканирования.

При работе с растровыми данными важной является задача определения местоположения ячейки в последовательном файле по растровым координатам и наоборот. Для обычного порядка сканирования и для Boustrophedon – сканирования получение такого отображения не составляет труда. При сканировании по Мортону задача усложняется.

Рассмотрим пример. Пусть требуется по растровым координатам ячейки A(2, 3) определить ее номер в последовательности Мортона. Для этого представим координаты A в двоичной системе счисления и на их основе сформи-

руем число N так, что координаты столбца ячейки A задают нечетные биты N, а координаты строки – четные биты. Получившееся число $N=(1\ 1\ 0\ 1)_2=13$ соответствует позиции ячейки A в последовательности Мортона.



21	22	25	26	37	38	41	42
20	23	24	27	36	39	40	43
19	18	29	28	35	34	45	44
16	17	30	31	32	33	46	47
15	12	11	10	53	52	51	48
14	13	8	9	54	55	50	49
1	2	7	6	57	56	61	62
0	3	4	5	58	59	60	63

AAAAABBBBBBAAABAA 5A 5B 3A 1B 2A

A JD JA ID ZA

Рис.	3.	Порядок	сканирования	растра	Пеано.
		- F / 1-	· · · · · ·	F F	

Обратная задача решается похожим способом. Пусть ячейка *В* записана в десятой позиции последовательности Мортона. Представим ее номер в двоичной системе счисления и разделим четные и нечетные биты между растровыми координатами столбца и строки ячейки: $N=10=(1010)_2$. Получим растровые координаты ячейки A(3, 0).

1.2. Иерархические структуры данных

Рассмотренные выше порядки сканирования растровых изображений дают незначительные различия в компрессии данных. Основное преимущество Мортон-сканирования и других иерархических структур данных заключается в более быстром доступе к данным. Информация распределена по карте неравномерно. Увеличение разрешения растрового изображения приводит к увеличению размеров файлов, а уменьшение – к потере информации. Далее пойдет речь об адаптивных методах представления растровых данных с разной плотностью информации.

На рис. 4 изображена растровая матрица размера 16 х 16, в которой содержатся 255 значений "А" и одно "В". Индексируем растр следующим способом. Разделим матрицу на четыре подматрицы размера 8 х 8 и нумеруем их 0, 1, 2, 3 в порядке Мортона. Назовем подматрицу гомогенной, если в ней содержатся одинаковые значения. Будем рекурсивно разбивать негомогенные подматрицы до тех пор, пока не достигнем гомогенности всех подматриц. Таким способом получим адаптивное разрешение растрового изображения, где участки с меньшей плотностью информации представлены крупными блоками ячеек, а с большей плотностью – мелкими блоками ячеек.

Идея выделения гомогенных блоков растра тождественна кодированию растра по Мортону. Гомогенный блок растра размера $m \ x \ m$ при сканировании по Мортону соответствует коду $\langle m^2, A \rangle$.



Рис. 4. Разбиение растра на гомогенные блоки.

Соответствие двумерных растровых координат ячейки и адреса ячейки в последовательном файле похоже на аналогичное преобразование при кодировании растра по Мортону. Единственное отличие в том, что используется система счисления с основанием четыре. В примере на рис. 4 ячейка "В" имеет код 0311. В двоичной системе счисления $N=0311=(00110101)_2$. Разделим биты между растровыми координатами и выясним, что ячейка лежит в четвертой строке и седьмом столбце.

Представленные таким способом растровые данные соответствуют квадродереву, вершина которого – исходное изображение, а листья – гомогенные блоки ячеек. При кодировании квадродеревьев ячейки на каждом уровне могут содержать либо значение гомогенного блока, либо указатель на следующий уровень. Дерево, показанное на рис. 4 может быть представлено в виде линейной последовательности следующим образом.

		$\overline{\mathbf{v}}$				$\overline{\mathbf{v}}$				$\overline{\mathbf{v}}$				$\overline{\mathbf{v}}$			
Позиция:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Содержание:	2	6	Α	А	Α	Α	Α	Α	10	Α	14	Α	Α	Α	В	Α	Α
Уровень:	0	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4

Рис. 5. Кодирование квадродерева.

Как уже отмечалось выше, основное преимущество иерархической организации данных в ГИС заключается в пространственном упорядочении информации и более быстром ее поиске. Поэтому рассматриваются две задачи ГИС, связанные с индексированием квадродеревьями: первая – поиск всех частей карты с заданным значением и вторая – определение содержимого некоторой ячейки.

Обозначим n – число уровней квадродерева (тогда размер растра 2n*2n) и через m – число листьев в дереве. Чтобы найти части карты с некоторым значением B, необходимо проверить каждый лист дерева, что потребует mшагов. Определение значения ячейки происходит путем спуска по квадродереву до тех пор, пока не будет получен гомогенный блок. В худшем случае, когда ячейка находится на самой вершине дерева (как, например, ячейка B на рис. 4), поиск займет n шагов. Сравним теперь трудоемкости обеих задач на квадродереве с трудоемкостями этих задач при различных вариантах сканирования растра.

Таблица 1.

Структура данных	Поиск частей с задан-	Определение значения
	ным значением	ячейки
Квадродерево	m	n
Обычный порядок	$4 n^2 *$	1 **
Boustrophedon	m ***	m ****
Мортон	m ***	m ****

Трудоемкость алгоритмов при различной организации растров.

Прим. * – проверяется каждая ячейка матрицы; ** – непосредственное вычисление позиции ячейки; *** – число цепочек примерно соответствует числу листьев; **** – проверяется каждая цепочка.

Существуют различные модификации квадродеревьев, позволяющие, например, эффективно индексировать трехмерные данные (при этом куб рекурсивно делится на восемь частей). При кодировании глобальных данных в проекции Меркатора, представленных в растровой форме, существует проблема различий в форме и размерах ячеек, что приводит к отклонениям в модели. Проблема может быть решена путем представления данных в иерархической форме. Для этого строится глобальная тесселяция: земная поверхность проецируется на октаэдр, содержащий восемь пронумерованных треугольников. Далее каждый треугольник делится на четыре треугольника соединением отрезками середин его сторон. Получившаяся модификация квадродерева позволяет получать разрешение 1 метр при уровне вложенности дерева, равном 20.

1.3. Алгоритмы на квадродеревьях

Иерархическая организация данных позволяет получать быстрые способы доступа к пространственным данным. Рассмотрим теперь некоторые алгоритмы ГИС на квадродеревьях: вычисление площади, оверлейный алгоритм и алгоритмы определения смежности ячеек.

Чтобы определить площадь ячеек с некоторым значением в растровом слое, необходимо обойти дерево и подсчитать количество ячеек, кодированных этим значением, взвешенное площадью ячейке на данном уровне дерева. Вычислим, например, на карте 1 (рис. 6) площадь ячеек со значением "A". Площадь $S_A = 1*(Count leaf (00,02,03,32)) + 4*(Count leaf (2)) = 8.$



Рис. 6. Оверлейная операция на квадродереве.

Оверлейная задача на квадродеревьях заключается в совмещении квадродеревьев двух карт и получении нового квадродерева, индексирующего обе карты. Для этого требуется одновременно обойти оба дерева, следуя ветвям, существующим в обоих деревьях. В тех узлах, где у одного из деревьев будет отсутствовать ветвление, значение атрибута переносится на все последующие подуровни. В результате образуется "широкое" дерево, содержащее оба атрибута (рис. 6). Многие операции ГИС, работающие с иерархическими структурами данных, требуют наличия способов определения смежности ячеек. Для этого будем использовать представление координат ячеек в системе счисления с основанием 4 и разделять биты так, как это делалось в сканировании растра по Мортону. При этом используется tesseral – арифметика, в которой перенос между разрядами осуществляется через две позиции. Например, разность 1000 - 1 = 0010, а сумма 1 + 0001 = 0100.

Будем различать два случая: когда проверяемые ячейки имеют коды одинаковой и разной длины. Блоки одинакового размера являются смежными, если их представления в tesseral – арифметике отличаются на 1 или 10. Например, блоки 01 и 03 являются смежными, так как 0011 - 0001 = 10. Блоки 033 и 211 также смежные, так как 001111 + 10 = 100101. Блоки 01 и 30 не являются смежными, так как 1100 - 0001 = 1001.

Для определения смежности блоков разного размера коды приводятся к основанию 2. Далее с кодом большей длины суммируются ± 01 и ± 10 . Из получившихся четырех кодов отбрасываются как невозможные все отрицательные (по переносу). Оставшиеся коды сдвигом вправо приводятся к длине кода меньшей длины. Два блока являются смежными, если трансформированный и обрезанный код большей длины равен короткому коду.

Например, требуется определить, являются ячейки 02 и 2 смежными. Приведем коды в двоичной системе: $02_4 = 0010_2$, $2_4 = 10_2$. Прибавим к длинному коду ±01 и ±10. Получим 0010+1=0001, 0010+10=1000, 0010-10=0000. Разность 0010-1 отрицательная, эта комбинация отбрасывается как невозможная. Выбрав два старших разряда из оставшихся результатов, получим коды $00_2=0_4$ и $10_2=2_4$. Один из получившихся кодов равен короткому коду, поэтому ячейки 2 и 02 являются смежными.

1.4. Пространственные индексы

В векторных ГИС пространственные индексы используются для более быстрого доступа к объектам на определенном участке карты. Индексирование пространственных объектов позволяет уменьшить вычислительную сложность процедур поиска пересекающихся и вложенных объектов, поэтому индексы являются важной частью алгоритмов оверлея полигонов.

Процесс построения индекса для цифровой карты включает следующие шаги. Сначала для каждого объекта базы данных находится наименьший лист квадродерева, полностью включающий объект. Некоторые крупные объекты могут лежать более чем в одном квадранте первого уровня квадродерева. В этом случае объекты помечаются значением "NULL". Остальные объекты помечаются кодом включающего листа квадродерева. Затем объекты сортируются по возрастанию получившегося ключа, а сам индексный файл в свою очередь индексируется обычным способом (рис. 7).

Построенные таким способом индексы используются для поиска объектов, пересекающих заданный полигон или линию. Для этого определяется

минимальный лист квадродерева, включающий заданный объект. Поднявшись из полученного узла до вершины дерева и выполнив обход поддерева, корнем которого является этот узел, получим список листьев дерева, внутри которых объекты могут пересекаться с заданным объектом.

Очевидно, пространственные индексы, построенные на квадродеревьях, более эффективны по сравнению с независимым упорядочиванием объектов по *x* и *y*, так как в этом случае учитывается пространственный характер данных. Индексирование квадродеревьями наиболее целесообразно для мелких объектов (особенно для точек). Большим объектам обычно соответствуют крупные блоки. Для них часто требуется определять пересечения с другими объектами.



Рис. 7. Индексирование цифровой карты квадродеревом.

Проблему индексации крупных объектов можно решить с использованием R-деревьев (R – rectangle, прямоугольник), в которых также используется концепция минимального вмещающего прямоугольника. Здесь требуется найти два прямоугольника таких, что внутри них расположено максимально возможное число объектов. При этом нужно стремиться, чтобы количество объектов в прямоугольника должно быть приблизительно одинаковым. Прямоугольники могут пересекаться, но площадь пересечения должна быть настолько малой, насколько возможно. Далее эта процедура рекурсивно повторяется (рис. 8).



Рис. 8. Индексирование цифровой карты R-деревом.

2. Алгоритмы вычислительной геометрии

В геоинформационных системах сложные алгоритмы анализа часто строятся из простых алгоритмов. Рассмотрим сначала некоторые простые алгоритмы, а далее покажем, как из этих простых алгоритмов строятся сложные аналитические процедуры.

2.1. Пересечение линий

Операция нахождение пересечения линий является одной из базовых в ГИС–анализе. Она используется в оверлейных операциях с полигонами, при соединении и разъединении (merge и dissolve) линий и полигонов. Эта операция является базисной при определении нахождения точки в полигоне, при удалении расщепленных полигонов. Поэтому эффективные алгоритмы определения пересечения линий важны в любой векторной ГИС.

Рассмотрим простейший пример: требуется определить, пересекается ли отрезок AB (4, 2) – (2, 0) с отрезком CD (0, 4) – (4, 0) и если да, то в какой точке? Для этого нужно найти уравнения прямых AB и CD и решить их совместно (рис. 9-а). Уравнение прямой y=a+bx может быть найдено по двум точкам, через которые она проходит. Коэффициент наклона прямой $b=(y_2-y_1) / (x_2-x_1)$. Используя любую из точек, через которые проходит прямая, найдем $a=y_i-bx_i$. Уравнение первой линии y=x-2, а второй линии y=4-x. Сложив два уравнения, получим точку пересечения (3, 1).



Рис. 9. Точка пересечения прямых: а) внутри отрезков; б) снаружи.

В общем виде две линии, заданные уравнениями $y=a_1+b_1x$ и $y=a_2+b_2x$, пересекаются в точке $x = -(a_1-a_2) / (b_1-b_2)$; $y = a_1+b_1x$. Однако таким способом можно найти только точку пересечения непараллельных линий бесконечной длины. Возможно отрезки не пересекаются, а пересекаются продолженные по этим отрезкам прямые (рис. 9-б). Отрезки пересекаются, если для точки пересечения (x, y) и точек A, B, C, D выполнены условия:

$$(x_A - x)(x - x_B) >= 0; \quad (x_C - x)(x - x_D) >= 0; (y_A - y)(y - y_B) >= 0; \quad (y_C - y)(y - y_D) >= 0.$$

Необходимо учитывать специальные случаи. Для вертикальных линий угол наклона b стремится к бесконечности, поэтому точку пересечения ищут особым способом. Если обе линии вертикальные, они не пересекаются. Если вертикальная только одна из линий, то подстановкой решается система уравнений y=const и $y=a_2+b_2x$. Невертикальные параллельные линии также вызывают сбой в работе алгоритма, поэтому перед решением системы уравнений следует проверять b_1-b_2 на равенство нулю.

Рассмотрим теперь способы определения пересечения полилиний. Пусть имеются две полилинии с n_1 и n_2 сегментами соответственно. Самым простым способом нахождения их точек пересечения является последовательная проверка пересечения каждого сегмента первой линии с каждым сегментом второй линии. Сложность этого алгоритма, пропорциональная произведению $n_1 * n_2$, может быть уменьшена при помощи разнообразных эвристических алгоритмов. Хотя в этих алгоритмах требуются дополнительные шаги обработки и, возможно, структуры данных, общая трудоемкость алгоритма снижается. Рассмотрим некоторые из таких методов.

Сложность алгоритма вычисления пересечения полилиний может быть снижена, если предварительно проверять на пересечение минимальные ограничивающие прямоугольники полилиний. Эти прямоугольники определяются минимальными и максимальными координатами x и y. Две полилинии не пересекаются, если не пересекаются их ограничивающие прямоугольники. Можно применить этот подход и для определения пересечения отдельных сегментов полилиний. Два отрезка AB и CD не пересекаются, если не пересекаются интервалы (x_A, x_B) и (x_C, x_D) или не пересекаются интервалы (y_A, y_B) и (y_C, y_D) .

Следующий метод, впервые использованный в ГИС ArcInfo, основан на разбиении полилинии на секции, в которых линия монотонно возрастает или убывает по х и по у (рис. 10-а). Разбиение происходит в точках локального минимума или максимума по х или по у. Горизонтальная или вертикальная линия пересекает такую секцию только в одной точке. Это дает возможность уменьшить трудоемкость алгоритма поиска пересечения полилиний. Если для двух секций найдена точка пересечения, не нужно проверять оставшиеся пары точек, т.к. это пересечение единственное при условии, что вторые производные в секциях не меняют знак. Это ограничение может быть разрешено либо разбиением секций в критических точках, либо полным перебором пар сегментов для таких секций. Модифицированный таким способом алгоритм в некоторых случаях позволяет получить вычислительную сложность порядка $O(n_1 + n_2)$.

На рис. 10-б представлены два различных случая пересечения секций. В одном случае секции пересекаются только в одной точке, в другом – в нескольких точках. Определим условия, при которых точка пересечения единственна. Если две секции одновременно возрастают или убывают по одному направлению, одна из них возрастает, а другая убывает по другому, то полилинии на этих секциях пересекаются не более чем в одной точке. На рис. 10-в серым цветом выделены условия, при которых можно применять вышеописанный метод оптимизации алгоритма поиска пересечений полилиний.

Если требуется найти точки пересечения большого числа полилиний, как, например, в оверлейной задаче, можно организовать пространственную индексацию полилиний. Наиболее часто в ГИС используются индексы на квадродеревьях. При такой индексации поиск пересечений ведется только для полилиний, у которых ветви квадродерева пересекаются.





2.2. Операции с полигонами

Перейдем теперь к операциям с полигонами, заданными последовательностью вершин. Рассмотрим задачу определения площади полигона. Чаще всего применяется алгоритм, основанный на разбиении многоугольника на трапеции, ограниченные линией сегмента полигона, перпендикулярами, опущенными из вершин сегмента на ось х, и осью х (рис. 11-а). Для сегмента, соединяющего вершины (x_A, y_A) и (x_B, y_B), площадь такой трапеции равна $S = (x_B - x_A)^* (y_B - y_A) / 2$.

Вычислим площади трапеций для всех сегментов полигона и просуммируем их. Для сегментов, у которых $x_i > x_{i+1}$, площадь берется отрицательной (рис. 11-б). Следует заметить, что полигон – замкнутая фигура, поэтому нужно учитывать сегмент, соединяющий последнюю вершину с первой.



Рис. 11. Вычисление площади полигона: а) исходная фигура; б) разбиение на трапеции.

Таким способом можно вычислить площади не только для выпуклых многоугольников, но и для вогнутых, а также для полигонов, имеющих дыры. Алгоритм непригоден для вычисления площадей полигонов, имеющих самопересечения границ. Для полигонов, оцифрованных против часовой стрелки, площадь получается отрицательной. Проблемы возникают также при отрицательных значениях координат *у* вершин полигона. В таком случае можно либо прибавить к координатам *у* достаточно большое число, либо опускать перпендикуляры из вершин на прямую *y=const*, где const меньше самой малой *у*-координаты в полигоне.

Если используется система координат с большими значениями x и y (например, в системе координат Гаусса-Крюгера в районе Красноярска действуют координаты x=6200000; y=16500000), то при многократном суммировании площадей трапеций будет накапливаться вычислительная ошибка. Относительная погрешность получается особенно высокой для малых полигонов. Эта проблема может быть решена линейным преобразованием полигона к новой системе координат, в которой не будет столь больших значений, и вычислением в ней площади. Далее вычисляется площадь в исходной системе.

В модели "дуга-узел" полигоны формируются из дуг. При этом кодируется расположение полигона относительно направления цифрования дуги. Очевидно, определяемую границей двух смежных полигонов площадь достаточно вычислить один раз. Затем для правых полигонов эта площадь суммируется со знаком "плюс", а для левых – со знаком "минус".

Рассмотрим следующую задачу, часто встречающуюся в процедурах ГИС-анализа. Для заданной точки A(u, v) и полигона $P=(x_i, y_i)_{i=1..n}$ требуется определить, находится точка внутри полигона или снаружи. Задача может быть решена с использованием топологических свойств полигонов. Из точки A проведем вертикальную линию $(x_i, y_i) - (x_i, \infty)$ и вычислим количество пересечений этой линии с сегментами границы полигона. Если это число нечетное, точка лежит внутри полигона. Если число четное – точка лежит вне полигона (рис. 12).



Точка	Полигон	Число пересечений
1	Α	1
1	В	2
1	C	0
2	Α	0
2	В	1
2	C	0
3	Α	2
3	В	2
3	С	1

Рис. 12. Схема определения принадлежности точки полигону.

Вертикальная линия x=u пересекает сегмент с концевыми точками (x_i, y_i) $u(x_{i+1}, y_{i+1})$ когда эти точки расположены по разную сторону от вертикальной линии. Уравнение прямой сегмента y=a+bx определяется по его концевым точкам, а пересечение сегмента с вертикальной прямой x=u находится в точке (u, a+b*u). Вертикальные сегменты границы являются для этого способа определения принадлежности точки полигону специальным случаем. Когда $x_i=x_{i+1}$ и $x_i <> u$, линия и сегмент границы не пересекаются. Когда $x_i=x_{i+1}$ и $x_i=u$, прибавим к количеству пересечений 0 или 2. Для сохранения детерминированности алгоритма нужно также проверять, расположена ли точка на границе полигона (a+b*u=v). Иначе алгоритм в разных случаях будет выдавать разные результаты (рис. 13).



Рис. 13. Неопределенность при расположении точки на границе объекта.

Если граница полигона разбита на монотонные секции, вертикальная линия пересекает секцию не более чем в одной точке. Поэтому когда найдено одно пересечение, можно не проверять оставшиеся сегменты секции на пересечение с вертикальной линией. Алгоритм можно применять также и вогнутых полигонов, для полигонов, имеющих дыры и самопересечения границ.

Перейдем к задаче определения центральной, репрезентативной точки полигона. В ГИС с этой целью часто используется понятие центроида – точки, являющейся центром тяжести полигона. Как видно из рис. 14, центроид расположен не всегда внутри полигона. Однако эта операция может применяться, например, в картографической генерализации при замене площадных объектов точечными.



Рис. 14. Расположение центроидов полигонов.

Координаты центроида региона с площадью А вычисляются по следующим формулам: $x_{u} = \Sigma ((y_{i} - y_{i+1}) (x_{i}^{2} + x_{i} x_{i+1} + x_{i+1}^{2}) / 6A); y_{u} = \Sigma ((x_{i} - x_{i+1}) (y_{i}^{2} + y_{i} y_{i+1} + y_{i+1}^{2}) / 6A).$ В некоторых программах координаты центроида вычисляются как среднее значение *x* и *y*.

Рассмотрим следующую полигональную операцию – вычисление скелета полигона. Скелет полигона определяется как сеть линий, построенная таким образом, что каждая точка сети расположена на равном расстоянии от ближайших двух сегментов границы полигона. При помощи этой операции можно определить оптимальные места для подписи полигона. Скелет получается путем "сжатия" полигона (рис. 15). Сегменты границы полигона сдвигаются внутрь полигона на равные расстояния, поэтому можно считать эту операцию обратной построению буфера.



Рис. 15. Схема получения скелета полигона.

Выпуклые вершины многоугольника сдвигаются внутрь в направлении биссектрисы угла, формируемого смежными с вершинами сегментами. Вогнутые участки в скелете полигона заменяются дугами окружности. Правильные многоугольники превращаются в точки, являющиеся центрами вписанных окружностей.

2.3. Оверлей полигонов

Простые алгоритмы, рассмотренные в предыдущих разделах, формируют базис для более сложных алгоритмов ГИС-анализа, таких, как оверлеи полигонов. Эта операция традиционно используется в ландшафтном планировании, где с целью управления использованием земель исследуются пространственные взаимосвязи между наложенными друг на друга географическими слоями.

Оверлеи полигонов изоморфны операциям теории множеств. Когда накладываются два полигона A и B, получается графическая интерпретация объединения или пересечения множеств A и B. На рис. 16 показаны шестнадцать возможных оверлейных операций с двумя полигонами, выраженные через объединение, пересечение и отрицание помеченных цифрами 1 ... 4 множеств.

На основе оверлейных операций строятся некоторые другие функции ГИС. При визуализации данных интерес представляют только объекты, попадающие в "окно" пользователя, а остальные объекты для ускорения отображения должны быть пропущены. Для этого на слои карты накладывается прямоугольник – экстент карты, вне которого объекты не отображаются. При построении буферов вокруг точек, сегментов полилиний и полигонов создаются круги и прямоугольники, которые впоследствии оверлейной операцией сливаются в один объект.

Оверлейные операции применяются при площадной интерполяции. Здесь требуется распределить некоторую величину, связанную с полигоном A, между пересечением $A \cap B$ и разностью A - B пропорционально их площади. При этом считается, что плотность этой величины по всему полигону постоянна.



Рис. 16. Связь оверлеев полигонов с операциями теории множеств.

Перейдем теперь к способам реализации оверлейных операций. Будем рассматривать наиболее распространенный в ГИС-анализе случай, когда накладываются два слоя с непересекающимися полигонами. Представим, что в одном из слоев содержатся "красные" полигоны, а в другом – "синие". Тогда задача заключается в поиске полигонов на комбинированном "фиолетовом" слое. Атрибуты этого слоя содержат конкатенацию характеристик "синих" и "красных" полигонов. Количество полигонов, получившихся в результате наложения слоев, заранее предсказать нельзя. Из пересекающихся "синего" и "красного" полигонов может получиться сколь угодно много "фиолетовых".

Чтобы получить оверлей двух полигонов, вначале необходимо вычислить все пересечения между их границами. На рис. 17 изображен "красный" полигон с атрибутом "А" (тонкие линии) и синий полигон с атрибутом "1" (толстые линии). Внешняя часть на обеих картах имеет атрибут "0". Каждый полигон представлен одной дугой, для каждой из них известно, с какой стороны расположен полигон. После вычисления пересечений этих дуг образуются шесть новых дуг и четыре новых полигона, наследовавших атрибуты 00, A0, A1, 01. Для новых дуг также известно, какие полигоны лежат справа и слева.

По таблице смежности получившихся дуг и полигонов можно сформировать любой из возможных шестнадцати полигонов, показанных на рис. 8.



Рис. 17. Оверлей полигонов в модели "дуга–узел": а) исходные объекты; б) вычисление пересечений дуг; в) метки смежности дуг и полигонов.

Рассмотрим более сложный пример (рис. 18). Здесь накладываются слой с объектом "1" и слой с тремя объектами "A", "B" и "C". Вычислим пересечения дуг объектов и получим метки правых и левых полигонов получившихся новых дуг. Как видно из рисунка, дуги 3, 6 и 8 не имеют пересечений с объектом "1". Определим метки для третьей дуги. Легко видеть, что метки внутри полигона могут передаваться от дуги к дуге. Правый полигон третьей дуги тот же, что правые полигоны второй и четвертой дуги – "A1". Левый полигон шестой дуги тот же, что левые полигоны второй и четвертой дуги – "A1". Для третьей дуги, как для части сети "красного" слоя, известна "красная" часть метки левого полигона. В результате получается метка "B1". Аналогично, метка правого полигона для шестой дуги будет "B1".



Рис. 18. Оверлей полигонов с непересекающимися границами.

Восьмая дуга не пересекается с границей полигона "1" и не является смежной с другими дугами сети "красного" слоя. Для изолированной дуги при помощи алгоритма "точка в полигоне" следует определить вмещающий полигон "синего" слоя. Получим для восьмой дуги правый полигон "C1" и левый полигон "01". Последний шаг оверлейного алгоритма заключается в формировании полигонов из новых дуг путем обхода полигона от дуги к дуге до тех пор, пока полигон не замкнется.

Точность представления координат сегментов в машинной форме более высока, чем погрешности оцифровки и векторизации. Поэтому при поиске пересечений сегментов полигона могут возникать ошибки, связанные с отсутствием сведений о топологической структуре объектов. На рис. 19-а и на рис 19-б показаны два различных случая пересечения линий.



Рис. 19. Проблемы поиска пересечения полигонов: а) пересекающиеся сегменты; б) ложные пересечения сегментов смежных полигонов; в) расщепленные полигоны.

Из рисунков видно, что в первом случае линии пересекаются на самом деле, а во втором случае пересекающиеся участки линий представляют одну и ту же границу. Необходимо составлять оверлейные алгоритмы таким образом, чтобы различать эти ситуации. Полигоны, образующиеся при оверлее двух полигонов с ошибочно векторизованными общими границами, называются расщепленными. Для двух полигонов, состоящих из n_1 и n_2 точек, может образоваться до $(2n_1n_2/(n_1+n_2) - 3)$ расщепленных полигонов.

Расщепленные полигоны могут быть устранены либо в процессе оверлейной операции, либо после ее выполнения. В большинстве коммерческих ГИС используется первый подход, заключающийся в "нечетком" представлении линий. При этом для каждой линии задается уровень толерантности, связанный с возникающей из-за ошибок векторизации неопределенностью геометрии линии. Поиск пересечений ведется для "полос", заданных самой линией и уровнем толерантности (рис. 20-а). Следует заметить, что определение пересечений для нечетких линий не является транзитивным.



Рис. 20. Удаление расщепленных полигонов: а) до оверлейной операции; б) после оверлейной операции.

Для устранения расщепленных полигонов после оверлейной операции необходимо определить критерии, по которым расщепленный полигон можно отличить от настоящего. Расщепленные полигоны обычно имеют небольшую площадь и вытянутую форму. Они чаще всего состоят из двух дуг. Расщепленные полигоны характеризуются также "перемежающимися" атрибутами. Если синяя дуга с атрибутами "1" и "2" накладывается на "красную" дугу с атрибутами "А" и "В", расщепленные полигоны будут иметь атрибуты "В1" и "А2" (рис. 20-б). Дуги расщепленных полигонов заканчиваются в четырехвалентных узлах, а в реальных полигонах валентность узлов обычно равна трем. Состоящий из двух дуг расщепленный полигон можно заменить одной дугой, проходящей через центр полигона.

3. Моделирование поверхностей

В отличие от цифровых представлений точечных, линейных и площадных объектов, трехмерные объекты требуют особых форм представления, т.к. их местоположение описывается не только двумерными, но и высотными координатами. К наиболее распространенному типу трехмерных объектов относится топографический рельеф земной поверхности. При помощи трехмерных объектов могут быть также смоделированы карты плотности населения, атмосферного давления, влажности и т.п. Однако трехмерные модели традиционно связывают с цифровыми моделями рельефа (digital elevation model - DEM). Далее мы будем рассматривать модели рельефа, подразумевая возможность моделирования и других непрерывных феноменов и явлений.

Цифровые модели рельефа позволяют по конечному набору выборочных точек определять возвышение, крутизну склона, направление ската в произвольной точке на местности. Возможно выявление особенностей местности – бассейнов рек, дренажных сетей, пиков, впадин и т.п. Такие модели широко применяются во многих процедурах ГИС-анализа: при выборе места строительства зданий и коммуникаций, в анализе дренажных сетей, в анализе видимости, при выборе маршрута движении по пересеченной местности. Особенно широко цифровые модели рельефа применяются в гидрологии.

Поверхности являются непрерывными феноменами в противоположность дискретным объектам, выражаемым точками, линиями и полигонами. Но существуют способы представления поверхностей, в которых используется конечное количество точек. Разные подходы к выбору узловых точек, в которых известно значение возвышения поверхности, определяют две наиболее распространенные модели данных. В геоинформационных системах поверхности обычно описываются при помощи растровых моделей и триангуляционных сетей. В растровых моделях выборочные точки расположены в узлах регулярной растровой решетки, а в триангуляционных сетях – располагаются нерегулярно так, чтобы наилучшим образом "обогнуть" поверхность (отсюда название – triangulated irregular networks – TIN).

При моделировании непрерывных поверхностей также являются важными вопросы оценки возвышения поверхности в произвольной ее точке. В растровых моделях используется билинейная интерполяция, а триангуляционных сетях возвышение определяется из уравнения плоскости, заданной вершинами треугольника.

В обеих моделях могут быть вычислены производные к поверхностям, из которых наиболее часто используются угол и экспозиция склона. Угол наклона поверхности в некоторой точке обычно измеряют в градусах или процентах. Плоские регионы имеют нулевую крутизну склонов. Чем круче горы, тем больше угол наклона. Экспозиция склона характеризует направление наибольшего угла наклона в некоторой точке поверхности. Следует помнить, что подобные измерения имеют смысл только для прямоугольных систем координат, а для системы координат широта/долгота результаты будут неточны (особенно в северных широтах).

Исходными данными для создания цифровых моделей рельефа могут быть данные полевых измерений, изолинии рельефа топографических карт, профили фотограмметрических стерео–изображений, гидрографические и гипсометрические карты и т.п. Обратная операция также возможна. На основе растровых DEM или триангуляционных сетей можно построить карты рельефа в изолиниях, рассчитать профили поверхности между заданными точками. Эффектным средством визуализации топографической поверхности являются трехмерные изображения.

3.1. Растровые цифровые модели местности

В случае, когда выборочные точки располагаются в узлах регулярной решетки, цифровая модель рельефа может быть построена при помощи растровой модели. Как известно, эта модель имеет преимущества перед объектными моделями в простоте алгоритмов обработки и анализа данных, обусловленной простотой организации данных. Растровые DEM являются самым простым способом представления топографических данных и широко распространены.

Чтобы оценить возвышение произвольной точки, нужно определить, лежит точка в каком-нибудь узле сети. Если так, то значение возвышения выбирается непосредственно из базы данных. В противном случае необходимо выбрать процедуру оценки возвышения по ближайшим узловым точкам. Как грубое приближение можно использовать высоту ближайшей узловой точки (рис. 21-а). При этом значения высоты будут изменяться скачкообразно.



Рис. 21. Оценка возвышения в произвольной точке: a) – по ближайшей узловой точке; б) – аппроксимация МНК; в) – билинейная интерполяция

Более гладкую поверхность можно получить, если аппроксимировать значения высоты в области, ограниченной четырьмя точками сети (рис. 21-б). При этом необходимо учитывать, что полученная методом наименьших квадратов поверхность не обязательно проходит через узлы решетки, следовательно, в полученной поверхности вдоль соединяющих узлы линий будут разрывы. Будем искать приближение в окрестности узлов в виде плоскости $Z(x, y) = c_0 + c_1 x + c_2 y$. Коэффициенты $c_0 + c_1 x + c_2 y$ находятся из СЛАУ.

$$\begin{bmatrix} \sum_{i} 1 & \sum_{i} x & \sum_{i} y \\ \sum_{i} x & \sum_{i} x^{2} & \sum_{i} xy \\ \sum_{i} y & \sum_{i} xy & \sum_{i} y^{2} \end{bmatrix} \begin{bmatrix} c_{0} \\ c_{1} \\ c_{2} \end{bmatrix} = \begin{bmatrix} \sum_{i} z \\ \sum_{i} zx \\ \sum_{i} zy \\ i zy \end{bmatrix}$$

Поверхность без разрывов (с разрывом первой производной) можно получить, используя билинейную интерполяцию (рис. 21-в). Выберем такую систему координат, что $x_1 = x_2$, $y_2 = y_3$, $x_3 - x_2 = 1$, $y_2 - y_1 = 1$. Найдем возвышения $Z_5 = Z_2 + (Z_3 - Z_2) x$ и $Z_6 = Z_1 + (Z_4 - Z_1) x$. Тогда $Z = Z_6 + (Z_5 - Z_6) y$.

Наклон и направление в произвольной точке растровой DEM вычисляются с использованием соседних точек. Обычно используется окно 3 х 3. Наклон поверхности определяется как отношение изменения возвышения к изменению горизонтального местоположения и выражается в процентах или градусах (рис. 22-а). Наклон измеряется в направлении наиболее крутого изменения возвышения. Чаще всего это направление не совпадает с направлением строк и столбцов растра (рис. 22-б). Для вычисления угла наклона будем использовать формулу $S_{epad} = \arctan \left[(\Delta Z / \Delta x)^2 + (\Delta Z / \Delta y)^2 \right]^{1/2}$. Направление склона (aspect) определяется как $A = \arctan \left[- (\Delta Z / \Delta y) / (\Delta Z / \Delta x) \right]$.



Рис. 22. Вычисление наклона поверхности: а) – вычисление угла наклона поверхности; б) – направление наиболее крутого изменения возвышения.

Рассмотрим следующий способ определения угла наклона поверхности в растровой ячейке DEM. Вычислим отношение $\Delta Z / \Delta x$ по значениям ячеек Z₄ и Z₅, а $\Delta Z / \Delta y$ – по ячейкам Z₄ и Z₅ (рис. 3-а). В этом примере расстояния между центроидами ячеек равно десяти, поэтому $\Delta Z/\Delta x = (49 - 40) / 20 = 0.45$; $\Delta Z/\Delta y = (45 - 48) / 20 = -0.15$. Отсюда угол наклона поверхности S_{град} = arctan [(0.45)² + (-0.15)²]^{1/2} = 25.38°. Вычислим направление склона: $A = \arctan[-(-0.15) / (0.45)] = 18.43°.$



Рис. 23. Вычисление угла наклона поверхности в растровой ячейке по четырем соседним: а) – схема вычисления; б) –ядра преобразования.

Вычислить угол наклона поверхности во всех ячейках растрового слоя можно при помощи двух преобразований, ядра которых приведены на рис. 23-б. Эти преобразования позволяют получить для ячейки значения $\Delta Z/\Delta x$ и $\Delta Z/\Delta y$, по которым вычисляется угол наклона. Исходный растровый слой m x n точек обрабатывается скользящим окном размера 3 x 3. В результате этой фокальной операции получается растровый слой, в ячейках которого содержатся значения угла наклона, и имеющий размер (m - 2) (n - 2).

Как видно из рис. 23-а, при определении угла наклона поверхности в растровой ячейке DEM по четырем соседним не используются угловые ячейки окна – z_1 , z_3 , z_6 , z_8 . Рассмотрим способ вычисления угла наклона поверхности по конечным разностям третьего порядка (рис. 24-а).



Рис. 24. Вычисление угла наклона поверхности в ячейке по конечным разностям 3-го порядка: а) – схема вычисления; б) –ядра преобразования.

Здесь $\Delta Z/\Delta x = [1*(47-42) + 2*(49-40) + 1*(52-44)] / 80 = 0.39; \Delta Z/\Delta y = [1*(47-52) + 2*(45-48) + 1*(42-44)] / 80 = -0.16. Отсюда угол наклона поверхности <math>S_{2pad} = \arctan [(0.39)^2 + (-0.16)^2]^{1/2} = 22.9^\circ$, а направление склона $A = \arctan [-(-0.16) / (0.39)] = 22.36^\circ$.

По растровому слою, содержащему углы наклона поверхности, может быть построена тематическая карта. Для этого задается легенда – шкала углов наклона с соответствующими цветами отображения. Пример такой тематической карты приведен на рис. 25-а. На физических картах с целью улучшения восприятия изображения рельеф показывается с отмывкой – пластическим полутоновым изображением рельефа путем наложения теней. На цифровых картах такой эффект можно получить, раскрашивая ячейки растровой DEM в соответствии с экспозицией склонов. Если источник освещения расположен на северо-западе, светлыми цветами раскрашиваются ячейки с экспозицией 270°–360°, темными цветами – 90°–180°, средними цветами – 0°–90° и 180°–270° (рис. 25-б).



Рис. 25. Карты рельефа: а) – тематическая карта углов наклона топографической поверхности; б) – изображение рельефа с отмывкой по экспозиции.

Растровые DEM могут использоваться в гидрологии для построения моделей стока, определения водоразделов рек (рис. 26). Направление стока из ячейки растра определяется ее высотой и высотой соседних ячеек. При этом могут рассматриваться все восемь соседей (движение королевы) или четыре соседа по вертикали и горизонтали (движение ладьи). Будем считать, что из данной ячейки имеется сток на соседнюю, если высота соседней ячейки меньше высоты других соседних ячеек и высоты данной ячейки.

В растровых DEM анализ видимости по вычислительной сложности значительно проще. Анализ видимости - операция обработки цифровых моделей рельефа, обеспечивающая оценку поверхности с точки зрения видимости или невидимости отдельных ее частей путем выделения зон и построения карт видимости с некоторой точки обзора или множества точек, заданных их положением в пространстве (рис. 27-а).



Рис. 26. Использование растровых DEM в гидрологии: а) - направление стока, водоразделы и дренажная сеть, рассчитанные на растровой сетке; б) – тематическая карта дренажной сети и водоразделов.

Пусть наблюдатель находится в ячейке с высотой 100 м. Справа от наблюдателя расположены ячейки с высотами { 110; 125; 115; 160; ... }. Легко видеть, что ячейки с высотами 110, 125, 160 будут наблюдателю видны, а ячейка с высотой 115 – нет. На рис. 27-б показана трехмерная модель с рассчитанными для заданного положения наблюдателя зонами видимости.



Рис. 27.Анализ видимости: a) – схема определения видимости; б) – пример зон видимости для заданного положения наблюдателя (белый цвет).

3.2. Нерегулярные триангуляционные сети (TIN)

Нерегулярные триангуляционные сети (Triangulation Irregular Network – TIN) являются альтернативой растровым DEM и используются во многих геоинформационных системах, системах автоматизированного картографирования, пакетах построения контуров. Модели TIN разработаны в 1970-х годах как простой способ построения поверхностей по нерегулярно расположенным точкам.

Модель TIN обладает некоторыми преимуществами перед растровыми DEM. В первую очередь, расположение точек адаптировано к местности: в равнинных участках точки расположены реже, а гористых – чаще. Выборочные точки соединяются прямыми отрезками, образующими треугольники, внутри которых поверхность задается плоскостью. Поверхность непрерывна, треугольники соединены между собой. Структуры данных в TIN-моделях более компактны и экономичны: TIN-модели из сотен точек может соответствовать растровая DEM из десятков тысяч точек. Несмотря не простоту модели, создание TIN требует решения ряда сложных задач: как размещать выборочные точки, как соединять точки в треугольники, как моделировать поверхность внутри треугольника.

Рассмотрим задачу выбора размещения точек триангуляции на следующем примере: имеется растровая DEM или оцифрованные изолинии рельефа, необходимо выбрать точки таким образом, чтобы наиболее точно представить поверхность в TIN-модели. Рассмотрим алгоритмы выбора точек DEM.

Алгоритм Фаулера – Литтла основан на поиске особых точек поверхности – пиков, хребтов, впадин и т.п. Поверхность проверяется скользящим окном размера 3 х 3. При этом соседи центральной ячейки помечаются плюсом, если их высота больше, и минусом если меньше. Очевидно, точка является пиком, если все восемь ее соседних ячеек помечены минусом. Аналогично, точка является впадиной, если все восемь ее соседних ячеек помечены плюсом. Точка является ущельем или перевалом, если плюсы и минусы вокруг точки образуют хотя бы два полных цикла:

 $\{++---++\} = 2$ цикла; $\{+-+-+-+-\} = 4$ цикла.

Далее слой обрабатывается окном 2 х 2 таким образом, что каждая ячейка по очереди становится во все позиции окна. Точка является гребнем горы, если в каждом из четырех окон она не самая низкая. Аналогично, точка принадлежит протоку, если во всех четырех окнах она не самая высокая. Затем, начиная от ущелий или впадин, ищутся связные протоки, пока не будут достигнуты пики (поиск можно вести и в обратном направлении). В результате получаются соединенные линиями пики, протоки, впадины, ущелья и перевалы. По выбранным точкам создаются треугольники. Алгоритм VIP (очень важная точка) в отличие от предыдущего алгоритма, в котором идентифицируются основные особенности местности, проверяет поверхность локально, используя окно 3 х 3. Это упрощение впервые использовано в ГИС ESRI Arc/Info. Ячейка в растровой DEM имеет восемь соседей, образующих четыре тройки (рис. 28-а).



Рис. 28. Алгоритм VIP: a) – четыре тройки ячеек; б) –расчет вариаций.

Для каждой тройки ячеек по соответствующей вариограмме рассчитывается коэффициент вариации (рис. 8-б). Первая тройка имеет нулевой коэффициент вариации, вторая и четвертая – низкий, а третья – высокий коэффициент вариации. Далее оценивается средняя вариация значения узла растровой DEM. Узлы с высокими показателями вариации включаются в результирующее разбиение, остальные отбрасываются.

Третий алгоритм выбора точек триангуляции основан на оптимизации существующего разбиения. Для заданной растровой DEM требуется найти такое подмножество точек (заданной мощности), что при соединении их линиями в треугольники получится как можно лучшее представление поверхности.

По узлам регулярной сети легко построить исходное разбиение на треугольники. В начале работы алгоритма разбиение полностью соответствует исходной растровой DEM. Далее все точки разбиения поочередно проверяются следующим способом. Точка временно удаляется, соответственно изменяется и разбиение. Затем определяются треугольники, содержащие удаляемую точку, оценивается разность возвышений этой точки, полученных из DEM и из трех верши треугольника. Эта разность записывается в базе данных и удаленная точка восстанавливается. Когда таким образом будут обработаны все точки, нужно удалить точки с наименьшими значениями запомненных в базе данных разностей.

После того, как выбрано необходимое количество узлов TIN, нужно выбрать способ разбиения поверхности на треугольники. При этом желательно получить близкие к равносторонним треугольники, чтобы произвольная точка поверхности была как можно ближе к узлам TIN, где значения возвышения известны точно. Рассмотрим два способа разбиения на треугольники. Триангуляция может быть получена путем упорядочивания точек по расстоянию между ними. Для этого вычисляются и сортируются по возрастанию расстояния между всеми парами точек. Ближайшие пары точек соединяются линией, если эта линия не пересекает полученные на предыдущих шагах линии. Процесс завершается, когда невозможно создать ни одной новой линии. В результате получится TIN-разбиение, в котором будет много остроугольных треугольников.

Этого недостатка лишена триангуляция Делоне. По определению три точки формируют треугольник в триангуляции Делоне тогда и только тогда, когда в окружности, описанной вокруг этого треугольника нет других точек разбиения. Разобьем поверхность на области, в которых каждая точка расположена ближе всего к некоторому узлу сети – генерирующей точке. Полученные границы называют полигонами Тиссена или полигонами Вороного. Две точки соединяются линией в триангуляции Делоне, если их полигоны Тиссена имеют общую границу. Этот метод позволяет получить требуемые "жирные" треугольники.



Рис. 29. Алгоритм Уотсона: a) – исходное разбиение; б) – выбор треугольников, описанная вокруг которых окружность содержит новую точку; в) – удаление треугольников; г) – разбиение выпуклого многоугольника.

Триангуляция Делоне может быть получена при помощи алгоритма Уотсона. В начале работы этого алгоритма создается супертреугольник, содержащий все точки разбиения. Точки последовательно добавляются в существующее разбиение. Опишем процедуру образования нового разбиения при добавлении новой точки (рис. 29). Сначала выбираются треугольники, описанная вокруг которых окружность содержит добавляемую точку. По определению эти треугольники не могут входить в триангуляцию Делоне, поэтому их следует удалить из разбиения. Выбранные треугольники разбиваются на отрезки, дублирующиеся отрезки удаляются. Оставшиеся отрезки формируют границу выпуклого многоугольника, который разбивается на новые треугольники простым соединением вершин с добавляемой точкой. По окончании работы алгоритма супертреугольник удаляется. Существует два основных способа хранения TIN: по треугольникам и по точкам. При кодировании сети по треугольникам для каждого треугольника в базе данных создается запись, содержащая его уникальный номер, координаты трех его вершин, а также номера трех смежных с ним треугольников (рис. 30-а). Во втором способе (рис. 30-б) для каждой точки разбиения сохраняется ее уникальный номер, координаты и список точек, с которыми она соединена прямыми (по часовой стрелке).



Рис. 30. Модели хранения TIN: a) – по треугольникам; б) – по точкам.

Рассмотрим способ вычисления наклона поверхности и экспозиции склона в TIN-модели. Для этого вычислим нормали к каждому треугольнику разбиения. Треугольник задается тремя точками (x_a, y_a) , (x_b, y_b) , (x_c, y_c) . Тогда нормаль Р к плоскости треугольника может быть выражена через векторное произведение двух его сторон как

$$P = \begin{bmatrix} (y_b - y_a) \cdot (z_c - z_a) - (z_b - z_a) \cdot (y_c - y_a) \\ (z_b - z_a) \cdot (x_c - x_a) - (x_b - x_a) \cdot (z_c - z_a) \\ (x_b - x_a) \cdot (y_c - y_a) - (y_b - y_a) \cdot (x_c - x_a) \end{bmatrix} = \begin{bmatrix} p_a \\ p_b \\ p_c \end{bmatrix}$$

Теперь вычислим направляющие косинусы и проекцию нормали Р к горизонтальной плоскости:

$$g = \frac{p_a}{\sqrt{p_a^2 + p_b^2 + p_c^2}}; \quad h = \frac{p_b}{\sqrt{p_a^2 + p_b^2 + p_c^2}}; \quad i = \frac{p_c}{\sqrt{p_a^2 + p_b^2 + p_c^2}}; \quad e = \frac{p_a}{\sqrt{p_a^2 + p_b^2}}; \quad f = \frac{p_b}{\sqrt{p_a^2 + p_b^2}};$$

После этого можно вычислить угол наклона поверхности треугольника α и экспозицию склона к северному направлению β (рис. 31):



Рис. 31. Графическое представление положения угла наклона и экспозиции склона треугольника.

Возвышение произвольной точки внутри треугольника определяется по уравнению плоскости, заданной вершинами треугольника. Плоскость с нормальным вектором (мы его уже вычисляли) $P = \{p_a, p_b, p_c\}$, проходящая через точку $M_0(x_0, y_0, z_0)$, описывается уравнением p_a ($x - x_0$) + p_b ($y - y_0$) + p_c ($z - z_0$) = 0. Отсюда по известным значениям х и у находятся возвышения произвольных точек.

Анализ стока в триангуляционных сетях осуществляется на основе рассчитанных для каждого треугольника угла наклона и экспозиции склона. При этом можно выделить два подхода к моделированию стока. В первом из них с треугольниками сети обращаются как с дискретными элементами: вода стекает с одного треугольника на другой в направлении максимального наклона. Второй способ предполагает работу с поверхностью как с мозаикой плоскостей треугольников. Здесь вода течет в треугольнике, собирается и движется вдоль дуг.

По растровым DEM и по триангуляционным сетям можно построить карты распределения характеристики по территории. Для этого нужно задать список значений z, по которым будут проведены изолинии. Например, можно взять максимальное и минимальное значения характеристики, разделить полученный интервал на несколько частей и с полученным шагом провести изолинии.

Изолинии заданной высоты строятся следующим способом. Рассмотрим все ячейки растровой DEM (или все треугольники триангуляционной сети). Будем искать пересечения ячеек (треугольников) с горизонтальной плоскостью Z=const. Из полученных отрезков нужно собрать полилинии. Для этого следует объединять смежные отрезки. Полилинии могут быть сглажены, например, при помощи сплайн–функций (рис. 32).



Рис. 32. Построение изолиний в растровой модели рельефа (а) и в триангуляционной сети (б).

Цифровые модели местности могут использоваться для построения профилей местности между выбранными точками. Для решения этой задачи нужно найти точки пересечения линии профиля с ячейками DEM или треугольниками TIN и оценить возвышение поверхности в полученных точках. Полученные точки соединяются и образуют полилинию: вдоль горизонтальной оси показывается протяженность профиля, а вдоль вертикальной – возвышения (рис. 33).



Рис. 33. Профиль топографической поверхности.

Литература

- 1. Варфоломеев И.В., Савельев А.С. Представление и обработка пространственных данных в ГИС: Методические указания / Красноярск: КГТУ, 2001.
- 2. Коновалова Н.В., Капралов Е.Г. Введение в ГИС. Петрозаводск: Изд-во Петрозаводского Госуниверситета, 1995.
- 3. Кошкарев А.В. Толковый мини-словарь основных терминов по геоинформатике (с английскими эквивалентами) // ГИС-обозрение, 1994. №1, с59-62.
- 4. Кошкарев А.В., Тикунов В.С. Геоинформатика / Под ред. Д.В. Лисицкого. М.: Картгеоцентр - Геоиздат, 1993.
- 5. Тикунов В.С. Моделирование в картографии: Учеб. М.: Изд-во МГУ, 1997.
- 6. Цикритзис Д., Лоховски Ф. Модели данных. М. Финансы и статистика, 1985.
- 7. Якубайлик О.Э Методы и приемы пространственного анализа в геоинформационных системах: Учебное пособие / Красноярск: Издательство КрасГУ, 2001.

Содержание

1. Структуры пространственных данных ГИС	6
1.1. Хранение растровых данных	6
1.2. Иерархические структуры данных	
1.3. Алгоритмы на квадродеревьях	
1.4. Пространственные индексы	
2. Алгоритмы вычислительной геометрии	
2.1. Пересечение линий	
2.2. Операции с полигонами	
2.3. Оверлей полигонов	
3. Моделирование поверхностей	
3.1. Растровые цифровые модели местности	
3.2. Нерегулярные триангуляционные сети (TIN)	
Содержание	