

УДК 681.323

ИНТЕГРАЛЬНЫЙ МЕТОД ВОССТАНОВЛЕНИЯ РЕЛЬЕФА ДОННОЙ ПОВЕРХНОСТИ ПО КАРТОГРАФИЧЕСКОЙ ИНФОРМАЦИИ

А.О.Пьявченко

Технологический институт Южного федерального
университета,
г.Таганрог

В статье рассматривается разработанный автором интегральный метод восстановления рельефа донной поверхности в трехмерный вид по неполному набору значений глубин заданного района плавания.

Предлагаемый к рассмотрению метод восстановления рельефа донной поверхности в трехмерный вид по картографической информации относится к классу волновых алгоритмов и основывается на анализе корреляционных связей между соседними точками растра. При этом метод базируется на известном рекурсивном методе пространственной фильтрации – методе пространственной свертки.

Метод позволяет проводить обработку двумерных карт поверхности морского дна, представленных значениями промерных глубин или линиями изобат. При этом анализируемая поверхность должна быть представлена с привязкой к узлам растра, причем исходные данные о поверхности (опорные точки) могут быть распределены по растру случайно, без всякой закономерности. Остальные точки обозначаются как неопределенные, то есть как раз такие, значения в которых необходимо вычислить в конце работы алгоритма реализации метода.

Шагу растровой сетки соответствует некоторое заранее определенное значение расстояния, отмеренное на плоскости, например, промерной карты по долготе или по широте между двумя точками донной поверхности. Следовательно, величиной шага можно регулировать исходный масштаб представления донной поверхности.

В ходе проводимых вычислений опорные точки не меняют до конца работы алгоритма своих координат и своего определенного заранее значения. Они также являются источниками волны, распространяющейся в четырех направлениях от анализируемой в данный момент времени опорной точки и инициируют вычислительный процесс по следующей схеме: вниз, вверх, влево и

вправо, что составляет одну итерацию в распространении волны. Состояние каждой точки на каждом направлении распространения волны просчитывается независимо от расчета состояния остальных точек. Причем, за одну четверть волны выполняется расчет всех точек раstra, не являющихся опорными. Это происходит каждый раз по мере формирования итерации волны и ее распространения во всех четырех направлениях. В итоге при растре 512x512 точек для худшего варианта представления исходных данных, когда определены только две опорные точки в противоположных углах сетки, получим искомый результат не более чем за 724 полные итерации распространения волны. Значение времени вычислений прямо пропорционально размерности самого раstra и обратно пропорционально числу изначально известных узловых опорных точек сетки.

Очевидно, что восстановленная таким образом донная поверхность имеет дискретное представление. При этом, чем больше шаг раstra, тем хуже точность восстановления поверхности из-за увеличения вероятности непопадания исходных координат в узлы сетки.

Рассмотрим предлагаемый метод более подробно на примере распространения четверть волны вниз (см. рисунок 1).

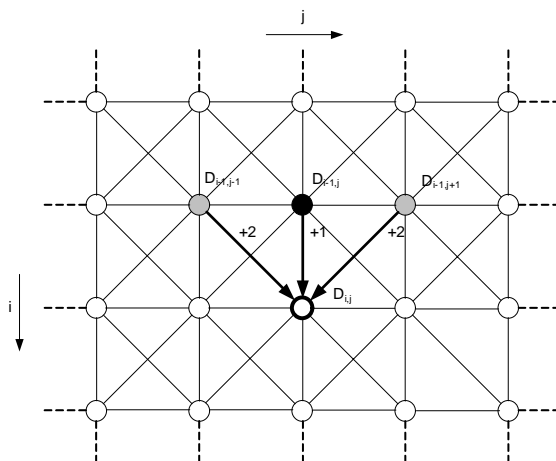


Рис. 1 - Один шаг распространения волны

Каждая узловая вершина сетки характеризуется двумя основными параметрами - расстоянием до ближайшей опорной точки $L_{i,j}$ и текущей глубиной $D_{i,j}$ в точке. На начальном этапе все точки раstra, кроме опорных, имеют нулевой код глубины, который принимается за код неопределенности. Код дистанции также принимается равным нулю для всех точек раstra.

В процессе расчетов дистанции до ближайшей опорной точки расстояние по вертикали и горизонтали между соседними узлами

сетки условно принято равным 1, а расстояние по диагоналям сетки - равным 2. Сделано это для сокращения используемой при расчетах разрядной сетки целочисленного вычислителя. Так, существуют модификации алгоритма, где расстояние между соседними узлами сетки равно 100 и 141, 1000 и 1414, соответственно, что, в свою очередь, позволяет увеличить точность производимых расчетов при соответствующем увеличении разрядной сетки вычислителя.

Анализируемая вершина (i,j) принимает “волну” значений глубин от соседних сверху вершин $(i-1,j)$, (i,j) , $(i+1,j)$.

Пусть хотя бы одна из указанных вершин имеет ненулевое значение глубины, а текущая расчетная вершина не является опорной. Напомним, что опорные точки в основной части алгоритма не пересчитываются.

Тогда текущее «кажущееся» значение глубины определяется по следующей формуле (1)

$$\tilde{D}_{i,j} = \frac{\sum C_i^p D_{i \text{ пред}}}{\sum C_i^p} = \frac{C_{i,j} \cdot D_{i-1,j} + C_{i,j-1} \cdot D_{i-1,j-1} + C_{i,j+1} \cdot D_{i-1,j+1}}{C_{i,j} + C_{i,j-1} + C_{i,j+1}}, \quad (1)$$

где $C_{i,j-1}$, $C_{i,j}$, $C_{i,j+1}$ - весовые коэффициенты, $D_{i-1,j-1}$, $D_{i-1,j}$, $D_{i-1,j+1}$ - значения глубин в соседних точках раstra на текущий момент времени, *не равные 0*.

Как уже отмечалось, данная процедура является разновидностью дискретной свертки, используемой в методах цифровой пространственной фильтрации. Основой алгоритмов фильтрации данного класса является одномерная или двумерная последовательность коэффициентов, называемая ядром свертки. Применяемый оператор строится на основании значений соседних точек, а также на последовательности особого вида констант или коэффициентов. В случае двумерных свертков рассматриваемый сигнал проходит обработку окном конечного размера и определенной формы (например, весь растр или меньше). Результатом работы является взвешенная сумма значений окрестных точек внутри рассматриваемого окна, где весом как раз и являются отдельные коэффициенты $C_{i,j}$.

В нашем методе каждый коэффициент $C_{i,j}$ рассчитывается как функция от значения кратчайшего расстояния L до ближайшей опорной точки от анализируемой с учетом принятых выше значений расстояний dL между соседними узлами сетки. Следует отметить, что в общем случае на выбор коэффициентов также оказывают влияние предельные значения величин перепадов глубин, изначально представленных в растре, а также требуемые точность и скорость

вычислений, однако этот вопрос не является темой обсуждения настоящей статьи.

В упрощенном варианте коэффициенты $C_{i,j}$ могут быть рассчитаны как функция, обратная дистанции по каждой дуге, то есть

$$C_{i,j+1} = LL_i - L_{i-1,j+1}^* + 1, C_{i,j} = LL_i - L_{i-1,j}^* + 1, C_{i,j-1} = LL_i - L_{i-1,j-1}^* + 1$$

где $L_{i-1,j-1}^*$, $L_{i-1,j}^*$, $L_{i-1,j+1}^*$ – значение кратчайшего расстояния до ближайшей опорной точки от точки (i,j) , LL_i – сумма значений кратчайших расстояний, рассчитанная по формуле:

$$LL_i = \sum_{k=-1}^1 L_{i-1,j+k}^*$$

$$\text{где } L_{i-1,j-1}^* = L_{i-1,j-1} + 2, L_{i-1,j}^* = L_{i-1,j} + 1, L_{i-1,j+1}^* = L_{i-1,j+1} + 2.$$

«Кажущееся» значение кода дистанции в точке (i,j) определяется как $\tilde{L}_{i,j} = \min(L_{i-1,j-1}^*, L_{i-1,j}^*, L_{i-1,j+1}^*)$.

После вычисления по формуле (1) «кажущегося» значения текущей глубины $\tilde{D}_{i,j}$ производится расчет истинной глубины обрабатываемой точки с округлением к ближайшему:

$$D_{i,j} = \frac{D_{i,j} + \tilde{D}_{i,j}}{2},$$

где $D_{i,j}$ – ненулевое значение глубины, полученное в узле на предыдущем шаге итерации (для правой части формулы).

Истинное значение кода дистанции определяется как результат нахождения минимума между его «кажущимся» и старым ненулевыми значениями для текущей точки раstra.

Если же первоначальное значение глубины $D_{i,j}$ все же равно 0, то его новое значение принимается равным $\tilde{D}_{i,j}$, а код дистанции его кажущемуся значению.

В соответствии с представленным выше методом можно выделить следующий алгоритм определения глубины в отдельно взятой точке раstra при одном проходе четверти волны сверху вниз.

1. Установка начальных значений $D_{i,j}$, $L_{i,j}$ для текущей точки раstra. При нулевом первоначальном значении глубины точка неопределенна, дистанция равна нулю, в противном случае это опорная точка с нулевым кодом дистанции.

Замечание 1. Все вычисления для опорной точки не производятся. Кроме того, рассматриваемая в статье версия алгоритма реализации метода не учитывает код береговой линии.

2. Расчет кодов дистанции $L_{i-1,j-1}^*$, $L_{i-1,j}^*$, $L_{i-1,j+1}^*$, а также их суммарного значения LL_i .

3. Подсчет значений весовых коэффициентов $C_{i,j-1}$, $C_{i,j}$, $C_{i,j+1}$.

4. Определение кажущихся значений глубины $\tilde{D}_{i,j}$ и дистанции $\tilde{L}_{i,j}$.
5. Расчет истинных значений глубины и дистанции анализируемой точки $D_{i,j}$ с учетом соответствующих предыдущих значений.
6. Повторение п.2-5 для следующего по схеме вычислений прохода (вверх, влево, вправо) с учетом замечания 1.
7. Повторение п.2-6 при распространении полной волны до границ растра.

На рисунках 2.а,...,2.г представлены результаты работы экспериментальной программы, реализующей вычисления согласно рассмотренному выше методу. Здесь показаны этапы исходного, начального представления поверхности в 2D-виде (рис. 2.а) для растра 64x64 и соответствующий ему трехмерный вид (рис. 2.б), затем результат вычисления первой волны (рис. 2.в) и окончательный вид полученной трехмерной поверхности (рис. 2.г).

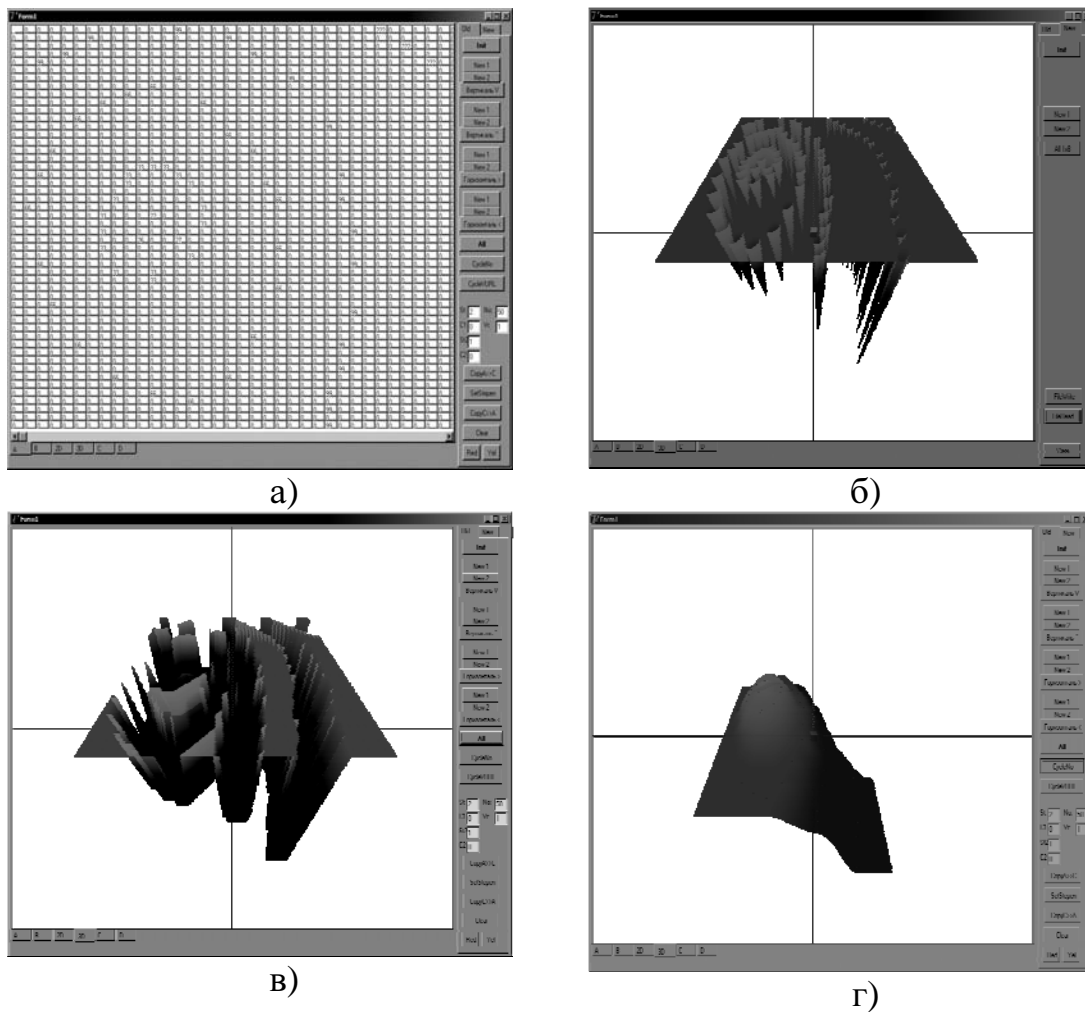


Рис. 2 - Результаты работы программы при расчете растра 64x64

В заключении необходимо отметить, что предложенный метод восстановления рельефа донной поверхности характеризуется естественным параллелизмом, векторизуемостью и масштабируемостью основных операций: алгоритм вычислений в каждом узле растра одинаков как для каждой четверти итерации, так и для всех итераций в целом. Это позволяет с высокой эффективностью выполнять решение задачи восстановления трехмерного рельефа на параллельных процессорах с SIMD архитектурой, например, на мультипроцессорах с конфигурацией связи типа n-куб, систолическая решетка и т.п.

В настоящее время на хоздоговорной основе коллективом кафедры вычислительной техники ТТИ ЮФУ проводятся работы по созданию макета параллельного вычислителя в базисе ПЛИС. Полученные в ходе выполнения работ результаты не только доказывают аппаратную реализуемость и эффективность метода, но и позволяют сделать необходимые оценки его характеристик, а также предложить различные его модификации.

Получено 01.06.07