

## Автоматизация ввода в компьютер изображений геофизических карт и построение их цифровых моделей

© В.И.Старостенко\*, В.В.Мацелло\*\*, И.Н.Аксак\*\*, В.А.Кулеш\*\*,  
О.В.Легостаева\*, Т.П.Егорова\*, 1997

\*Институт геофизики НАН Украины, Киев, Украина;

\*\*Институт кибернетики НАН Украины, Киев, Украина

Поступила 29 августа 1996 г.

Представлено членом редколлегии Е.Г.Булахом

Описано процедуру, алгоритмы і програмні комплекси вводу зображень геофізичних карт в комп'ютер за допомогою дигітайзера і сканера. Викладено алгоритм побудови цифрових моделей карт по введеному в комп'ютер їх зображенню. Наведено результати випробування алгоритмів (програмних комплексів) на модельних прикладах. Показано, що у випадку інформаційно простих карт ввід їх в комп'ютер за допомогою сканера виконується швидше, ніж за допомогою дигітайзера. Точність побудови цифрових моделей в обох випадках практично однакова. Під час роботи з інформаційно насиченими картами використанню сканера надається безумовна перевага.

The paper describes the procedure, algorithms and software for geophysical map's image inputting in computer using a digitizer and a scanner. Algorithm for constructing the digital model of map on it's image, entered in computer, is offered. The efficiency of elaborated algorithms (set of programs) is demonstrated on model tests. It is shown that information all simple maps are faster input in computer by the scanner than by the digitizer; the accuracy of constructing the digital models for both cases is identical. In the case of maps with complex information the use of the scanner has undoubted advantages.

**ВВЕДЕНИЕ.** Для обработки и интерпретации геофизической информации, представленной в виде карт, необходима автоматизация их ввода в компьютер с последующим определением по введенным данным значения поля в узлах заданной (обычно прямоугольной) сети. Такое матричное представление информации, отнесенной к некоторой постоянной отметке по высоте, называют числовой (цифровой) моделью геофизического поля [1, с. 85]. Сама же процедура ручного ввода карт в компьютер настолько громоздка, рутинна и столь необходима, что вопрос ее автоматизации был поставлен (и до определенной степени решен) еще на начальном этапе создания автоматизированных систем в геофизике [1, с. 72]. С появлением серийных специализированных устройств ввода графической информации в компьютер — дигитайзеров и сканеров [2] — появилась возможность решить проблему максимально полно, тем более что за последние годы быстро развивались теория и математическое обеспечение решения основных задач обработки изображений [3—8].

В настоящей статье описаны: 1) процедура и алгоритм (программа) ввода карт в компьютер с помощью дигитайзера, 2) процедура и алгоритмы (комплекс программ) ввода карт

в компьютер с помощью сканера, 3) алгоритм построения цифровой модели карты по введеному в компьютер с помощью дигитайзера или сканера ее изображению, 4) результаты опробования алгоритмов (программ) на модельных примерах.

Уточним, что геофизическая карта обычно содержит рамку с нанесенными на ней координатными метками, изолинии значений картируемой функции и цифровые надписи, которые обозначают их значение (рис. 1). В последующем принимается, что картируется гладкая непрерывная функция (вне зависимости от ее происхождения).

### ВВОД КАРТ С ПОМОЩЬЮ ДИГИТАЙЗЕРА.

Оператор с помощью курсора мыши дигитайзера в произвольном порядке обводит изолинии карты, предварительно размещенной на его планшете. В связи с большой разрешающей способностью дигитайзера (2725 × 1300 точек на 30,5 × 30,5 см) алгоритм (программа) предусматривает запись в выходной файл только необходимых координат точек (например, каждой  $i$ -й точки). Перед вводом изолинии оператору необходимо указать значение ее уровня. Если изолиния введена неточно (оператор следит за ее вводом на экране монитора), возможна корректировка изолинии, которая введена последней. Таким образом,

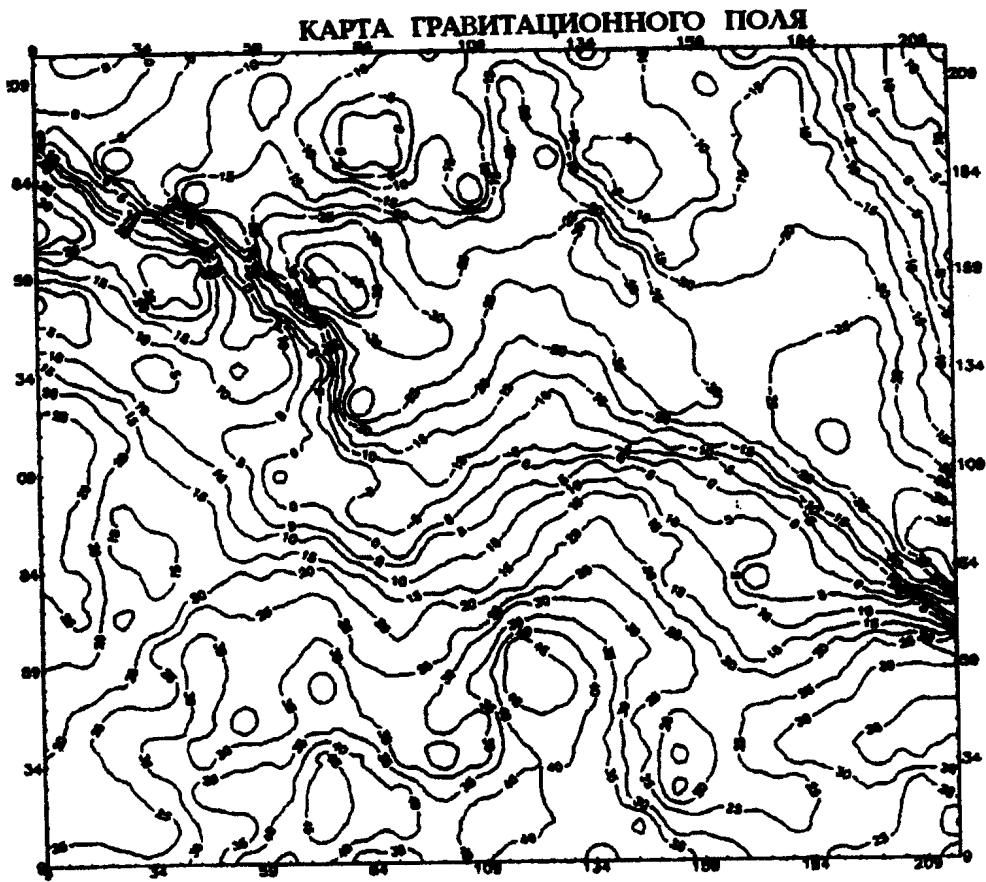


Рис. 1. Пример карты гравитационного поля.

точка за точкой формируются координаты отрезков, набором которых представляется изолиния. В результате ввода всех изолиний карты формируется окончательный выходной файл с полной исходной информацией. Она затем используется для построения цифровой модели карты.

Технология реализована в виде комплекса программ под управлением MS Windows, на алгоритмическом языке C согласно принципам модульности и структурирования. Модули запрограммированы нисходящим образом, т. е. первым дан головной модуль, затем подчиненные. Используемые принципы существенно облегчают тестирование и модификацию программы.

**ВВОД КАРТ С ПОМОЩЬЮ СКАНЕРА.** Ввод карт с помощью дигитайзера является трудоемкой операцией, требующей от оператора постоянного напряженного внимания, что приводит к ошибкам и необходимости неоднократной проверки и коррекции результатов кодирования. Ниже описан разработанный комплекс алгоритмов и программ, выполняю-

щий все наиболее трудоемкие операции автоматически под управлением оператора.

Программный комплекс состоит из двух основных частей. Первая обеспечивает ввод в компьютер растрового изображения, его удобный просмотр, предварительную обработку и преобразование в векторную форму. На основании результатов работы первой части вторая часть комплекса выполняет собственно кодирование карты изолиний, т. е. определение координат их точек. Рассмотрим подробнее работу программного комплекса.

**1. Предварительная обработка изображений.** В комплексе реализованы две основные научные идеи, позволяющие без использования специализированной аппаратной поддержки достигнуть уникальной скорости обработки графических изображений. Во-первых, это эффективный метод экономного кодирования бинарных (черно-белых) изображений, и, во-вторых, оригинальные быстродействующие алгоритмы обработки и векторизации, работающие на закодированных изображениях.

Первая часть комплекса программ выполняет следующие действия.

**1.1. Ввод и экономное кодирование изображения.** Программа ввода выполняет чтение растрового изображения карты из файла, представленного в формате TIFF, который поддерживается матобеспечением большинства современных сканеров.

Недостатком обычного растрового представления изображения является его большой объем. Поэтому операции обработки растровых изображений требуют больших объемов вычислений и, следовательно, работают медленно. Для сокращения компьютерной памяти, необходимой для хранения изображения, выполняется его экономное кодирование одновременно с вводом. Оно выполняется следующим образом.

Пусть  $T = \{(i, j) : 1 \leq i \leq m, 1 \leq j \leq n\}$  — прямоугольная решетка растра размером  $m \times n$  пикселей. Растровое графическое изображение есть функция  $v : T \rightarrow \{0, 1\}$ , где  $v(i, j)$  — цвет изображения в точке  $(i, j)$  черный, если  $v(i, j) = 0$ , либо белый, если  $v(i, j) = 1$ . Точка  $(i, j)$  является особой, если участок из четырех соседних точек растра  $(v(i, j), v(i + 1, j), v(i, j + 1), v(i + 1, j + 1))$  содержит нечетное количество черных точек. Приведем неформальное определение особой точки. Пусть отрезком контура называется граница между двумя соседними пикселями растра, причем один из пикселей черный, а другой — белый. Очевидно, контуры изображения состоят из вертикальных и горизонтальных отрезков. Точка (узел решетки растра) называется *особой*, если в ней отрезок контура изменяет направление с горизонтального на вертикальное или наоборот. Экономное кодирование изображения заключается в запоминании списка особых точек. Доказано, что на ограниченном растре преобразование из растрового вида в список особых точек является обратимым [4]. Таким образом, список особых точек содержит всю без исключения растровую информацию, однако количество особых точек для графических изображений значительно меньше количества точек растра.

В комплексе все операции обработки основаны на списке особых точек. Поэтому их трудоемкость пропорциональна не размеру растра, а количеству особых точек, которое значительно меньше.

**1.2. Сглаживание контуров.** В процессе сканирования решение о том, является точка ра-

стра белой либо черной, принимается в зависимости от того, превышает ли суммарное количество отраженного света в поле зрения светочувствительного элемента заданный порог. В случае, когда это поле зрения попадает на границу черного и белого участка изображения, количество отраженного света практически бывает случайным и может колебаться вблизи порогового значения. Это приводит к так называемому эффекту "бахромы".

Сглаживание "бахромы" приводит к чисто визуальному улучшению изображения, а также обычно позволяет значительно сократить объем списка особых точек. В комплексе реализован алгоритм сглаживания, который заключается в сдвиге границ между черными и белыми участками растра на одну клетку таким образом, чтобы количество особых точек стало как можно меньшим.

**1.3. Устранение помех.** Подмножество клеток растра  $T' \subset T$  на изображении  $v$  признается помехой, если:

- а) множество  $T'$  является связным;
- б)  $\forall t \in T'(v(t) = 0) \vee \forall t \in T'(v(t) = 1)$ ;
- в) не существует множества  $T'' \supset T'$ , для которого выполняется а) и б);
- г) размер минимального прямоугольника, который имеет вертикальные и горизонтальные стороны и охватывает  $T'$ , не превышает заданный размер.

Устранение помех заключается в замене значений  $v(t)$  для всех  $t \in T'$  на противоположные. В комплексе реализован быстродействующий алгоритм выделения связных областей изображения, измерения их геометрических размеров и удаления помех для изображений, представленных в виде списка особых точек [5].

**1.4. Устранение разрывов и тонких линий.** Назовем расширением изображения на  $k$  пикселей операцию преобразования изображения  $v$  в изображение  $w$  такое, что  $w(i, j) = 1$  тогда и только тогда, когда хотя бы для одной точки  $(i^*, j^*)$  из множества  $\{(i^*, j^*) : i^* \leq i \leq i + k; j^* \leq j \leq j + k\}$  выполняется  $v(i^*, j^*) = 1$ . Сужением изображения на  $k$  пикселей — это операция получения изображения  $w$ , для которого  $w(i, j) = 0$  тогда и только тогда, когда хотя бы для одной точки  $(i^*, j^*)$  из множества  $\{(i^*, j^*) : i - k \leq i^* \leq i; j - k \leq j^* \leq j\}$  выполняется  $v(i^*, j^*) = 0$ . Неформально говоря, сжатие на  $k$  пикселей заключается в  $k$ -кратном удалении ("соскабливании") слоя граничных черных точек, т. е. таких, у которых имеется хотя бы одна

соседняя белая точка. Соответственно расширение на  $k$  пикселей есть  $k$ -кратное наращивание слоя черных клеток, т. е. зачернение тех белых клеток, которые имеют хотя бы одну соседнюю черную клетку.

Устранение разрывов размером  $2k$  клеток выполняется как расширение на  $k$  пикселей с последующим сжатием на  $k$  пикселей. При этом при расширении некоторые близко расположенные черные области сливаются, а затем путем сжатия восстанавливается их первоначальная толщина.

Удалить тонкие линии толщиной, меньшей  $2k$  клеток, можно, последовательно сжав изображение на  $k$  пикселей, а затем расширив его на  $k$  пикселей.

В комплексе реализован алгоритм расширения — сжатия, трудоемкость которого не зависит от величины  $k$ . Идея алгоритма изложена в работе [6].

**1.5. Выделение скелета изображения.** Растровое изображение представляет собой матрицу, состоящую из нулей и единиц. Для пользователя, не являющегося специалистом по обработке изображений, более естественным и понятным является описание изображения в виде линий. Для получения такого описания используются процедуры векторизации или выделения скелета изображения. Алгоритмы скелетизации, использующие обычное растровое представление изображения, являются исключительно трудоемкими и работают очень медленно. В комплексе реализован быстродействующий алгоритм скелетизации [7], использующий список особых точек, основанный на понятии *нерасширяющегося (максимального) квадрата*. Квадратная область раstra называется нерасширяющимся квадратом, если все точки этой области черные, а любой другой квадрат, содержащий в себе данный, имеет хотя бы одну белую точку. Множество центров нерасширяющихся квадратов определяет совокупность прямолинейных отрезков, составляющих скелет изображения.

**1.6. Кусочно-линейная аппроксимация векторизованного изображения.** В результате погрешностей рисования, сканирования и скелетизации полученное векторное изображение может состоять из большого количества коротких отрезков, хотя визуально исходное изображение содержит длинные прямые линии. Для устранения этого недостатка, а также для уменьшения объема векторизованного изображения применяется кусочно-линейная

аппроксимация. Смысл ее заключается в том, что некоторый участок ломаной линии, являющийся частью скелетного изображения и состоящий из целого числа отрезков, заменяется одним спрямляющим отрезком, соединяющим концы этой ломаной, если расстояние от любой точки участка ломаной до спрямляющего отрезка не превышает заданного порога [8]. Для упрощения вычислений в отличие от обычной евклидовой метрики в программе в качестве расстояния между двумя точками используется минимальная сторона квадрата, содержащего эти точки.

**2. Кодирование карты изолиний.** Скелетное изображение, полученное в результате предварительной обработки и векторизации, представляет собой неориентированный плоский граф, вершинами которого являются концы прямолинейных отрезков скелета, а дугами — сами отрезки. Вершины графа имеют фиксированные координаты на плоскости.

Массив скелетного изображения — это набор некоторых списков. Каждый список соответствует одной вершине графа и выглядит так:

$$N, x, y, W, adr_1, adr_2, \dots, adr_N,$$

где  $N$  — количество отрезков скелета (ребер графа), выходящих из данной вершины;  $x, y$  — координаты вершины на плоскости изображения;  $W$  — длина стороны максимального квадрата в данной точке растрового изображения;  $adr_i, i = 1, N$  — адрес (далее ссылка) в скелетном массиве начала списка, соответствующего вершине, соединенной с данной вершиной дугой (отрезком скелета).

Для удобства предусмотрено совместное высвечивание растрового и скелетного изображения. Процесс кодирования карты изолиний заключается в последовательном выполнении следующих операций.

**2.1. Обнаружение отрезков изолиний.** На карте изолиний изображены два основных типа объектов: собственно изолинии и цифровые обозначения. Отличительным признаком этих типов является то, что изолинии представляют собой длинные неразветвленные ломаные, а цифры состоят из относительно коротких отрезков. Поэтому автоматическое обнаружение отрезков изолиний заключается в поиске длинных неразветвленных ломаных на скелетном изображении и их пометке. При этом у оператора есть возможность изменить результаты поиска, т. е. поме-

тить необнаруженные участки изолиний или отменить ложную отметку.

**2.2. Обнаружение рамки и очистка изображения вне рамки.** Изображение карты изолиний содержит еще один специфический класс объектов — рамку с нанесенными на ней координатными метками и дополнительную информацию, расположенную вне рамки (см. рис. 1). Координатные метки необходимы для сшивания результатов обработки различных листов карты. В данном комплексе они используются для вычисления параметров поворота изображения с целью компенсации возможного неправильного расположения исходной картинка в поле зрения сканера. Положение рамки используется для приведения результатов кодирования к системе координат, начало которой находится в левом верхнем углу рамки, а оси направлены вдоль верхней и левой сторон рамки.

Для выделения рамки оператор должен указать четыре точки скелетного изображения, соответствующие углам рамки. На скелетном изображении рамка представляет собой совокупность ломаных линий, соединяющих угловые точки. Обнаружение этих ломаных выполняется как поиск кратчайшего пути на плоском неориентированном графе. Для этого используется метод динамического программирования [9] и евклидова метрика. Если в результате искажения изображения (разрыв линий рамки) такой путь не был найден, оператору предлагается надлежащим образом отредактировать изображение и дополнительно указать необходимые точки.

Далее оператор должен указать восемь опорных точек (по две на каждой стороне рамки), необходимых для вычисления параметров поворота изображения. В качестве опорных точек выбираются координатные отметки, причем необходимо задавать на противоположных сторонах рамки строго соответствующие отметки, имеющие одинаковые глобальные координаты  $X$  на горизонтальных и одинаковые координаты  $Y$  на вертикальных сторонах.

После того, как введены опорные точки и выделена рамка, информацию об отрезках скелета, лежащих вне рамки, можно удалить. Это выполняется с помощью следующей процедуры. Строится четырехугольник, вершины которого — углы рамки, указанные оператором. Вблизи геометрического центра четырехугольника выбирается произвольная вершина графа. Так как четырехугольник выпуклый,

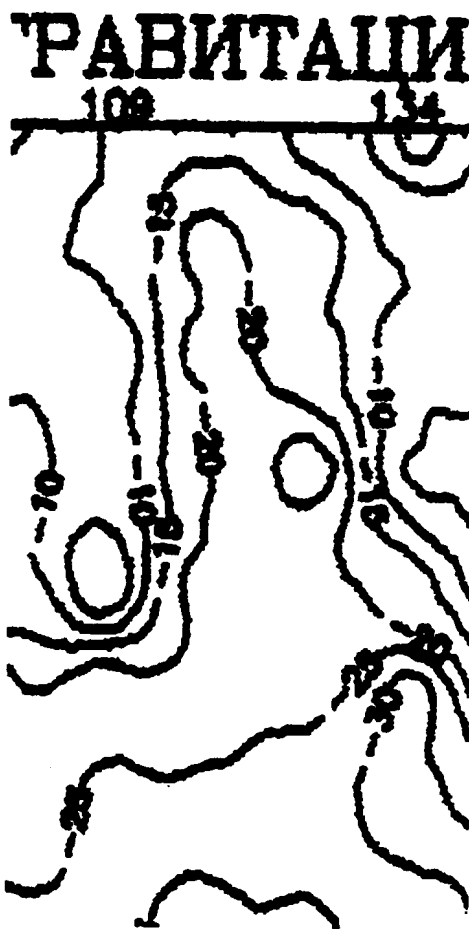


Рис. 2. Участок карты рис. 1, содержащий изолинии, цифровые обозначения, фрагмент рамки и внерамочной надписи.

то эта вершина обязательно лежит внутри рамки. После этого каждая вершина графа соединяется отрезком с выбранной вершиной. Если данный отрезок пересекает одну из сторон прямоугольника, то соответствующая вершина лежит вне рамки и, следовательно, сама вершина и выходящие из нее дуги удаляются из изображения.

**2.3. Прослеживание фрагментов изолиний.** Обычно изолинии и соответствующие им цифровые обозначения располагаются на карте таким образом: реальная изолиния разрывается на своем наиболее прямом участке (чтобы проще было доопределить изолинию на разрыве), и в этот разрыв вносится цифровое обозначение (рис. 2).

Естественно, при этом подразумевается, что изолиния проходит через цифровое обозначение, и легко на этом участке доопределяется (например отрезком прямой). Очевидно,

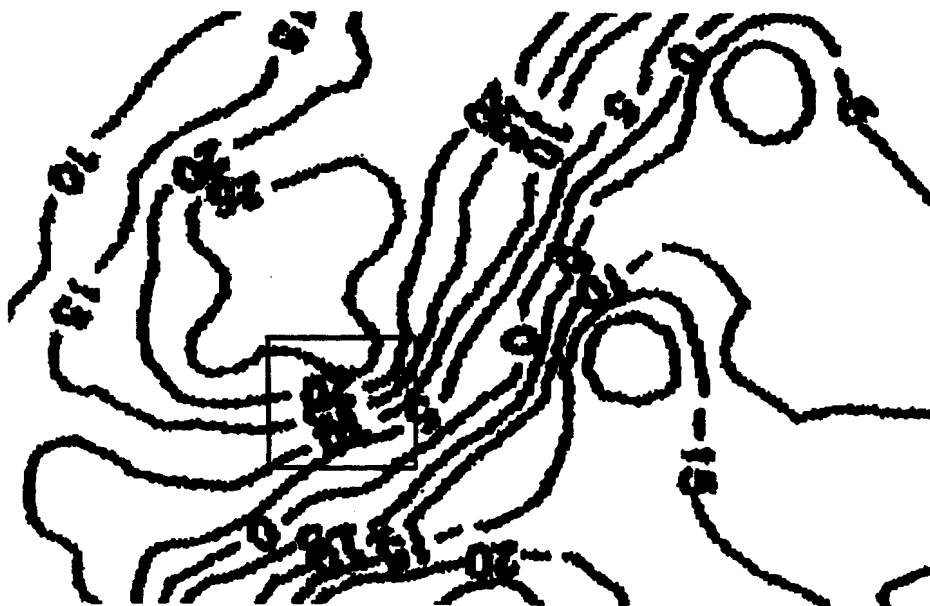


Рис. 3. Пример изображения информационно насыщенного участка карты.

при оцифровке карты фрагменты изолинии должны быть прослежены и собраны в одну изолинию. При этом должны выполняться два условия: 1) оба конца всех фрагментов изолиний должны соединиться либо с концами других фрагментов, либо с рамкой; 2) изолинии не должны пересекаться. Алгоритм, выполняющий такое прослеживание, должен рассортировать множество всех фрагментов изолиний на непересекающиеся подмножества, каждое из которых состоит из фрагментов, принадлежащих одной изолинии, и затем соединить фрагменты одной изолинии в непрерывную линию. Простейший метод — соединять ближайшие концы фрагментов изолиний — не решит задачу для информационно насыщенных участков карты (рис. 3), так как для правильного соединения фрагментов изолиний необходимо учитывать не только близость концов изолиний, но анализировать некоторую (иногда довольно большую) область карты в целом<sup>1</sup>.

Так оператор, прослеживая отдельную изолинию, к примеру на участке, обведенном квадратом рис. 3, прибегает к рассуждениям типа "конец этого фрагмента изолинии не может быть соединен с тем концом изолинии, так как в этом случае третий фрагмент не мо-

жет быть соединен ни с каким другим". Для разработки алгоритма решения этой задачи была предпринята попытка создать некоторую формализацию такого рода рассуждений.

**2.3.1. Основные определения. Определение 1.** Концом изолинии назовем ту точку скелетного графа, из которой выходит только одна дуга.

**Определение 2.** Фазой конца изолинии назовем угол между дугой, выходящей из этого конца, и осью  $X$ .

**Определение 3.** Связкой назовем несуществующую дугу, соединяющую два конца изолиний, если: 1) расстояние между этими концами меньше определенного порогового значения, 2) разность фаз этих концов лежит в заданных пределах, 3) эта дуга не пересекает ни одну скелетную дугу.

Будем считать скелетный массив фрагментов изолиний *допустимым* в смысле возможности прослеживания и в рамках приведенных выше определений, если существует такое множество связок  $S_d$ , что: 1) для каждого конца изолинии существует одна связка, 2) ни одна связка не пересекает другую связку.

Примем допустимый скелетный массив фрагментов изолиний *однозначно прослеживаемым*, если такое множество связок  $S_d$  единственно.

**2.3.2. Постановка задачи.** Пусть фрагменты изолиний представлены скелетным массивом. Необходимо сделать заключение о том, явля-

<sup>1</sup> Мысль о том, что для правильного суждения о положении изолинии необходим обзор фрагмента карты, по существу содержится в давно опубликованной работе [10].

ется ли скелетный массив допустимым и, если это так, то найти множество связей  $S_d$ . Если такое множество связей  $S_d$  найдено, то для прослеживания изолиний остается только внести связки в скелетный массив.

Примененный в описываемом программном комплексе алгоритм находит множество связей  $S_d$ , если скелетный массив является однозначно прослеживаемым. Однако ценность алгоритма была бы не велика, если бы он просто устанавливал этот факт, но не давал никаких разумных указаний по коррекции скелетного массива в случае, когда допустимое множество  $S_d$  отсутствует. Поэтому, если скелетный массив не является однозначно прослеживаемым, то алгоритм предъявляет оператору конец изолинии, для которого не может быть найдена подходящая связка и для которого, следовательно, необходима коррекция. Под коррекцией понимается возможность соединить два конца изолиний, добавить, убрать участок изолинии и т. д.

**2.3.3. Краткое описание алгоритма.** На входе алгоритма имеется скелетный массив фрагментов изолиний. Алгоритм состоит из последовательного выполнения следующих шагов.

**Шаг 1.** По скелетному массиву фрагментов изолиний строится список концов изолиний  $S_{ends}$ . Элемент этого списка  $S_{ends}(i)$  есть тройка  $(adr_i, \varphi_i, count_i)$ , где  $adr_i$  — ссылка в скелетном массиве на соответствующую концевую точку скелетного графа,  $\varphi_i$  — фаза соответствующего конца изолинии,  $count_i$  — счетчик количества связей, произошедших от данного конца изолинии. Первоначально поле  $count_i$  обнуляется. Его значение модифицируется при выполнении следующих шагов алгоритма.

**Шаг 2.** Над элементами списка концов изолиний  $S_{ends}$  проводится процедура построения массива всех возможных связей  $M_{conn}$ . Просматриваются все пары концов изолиний. Между  $i$ -м и  $j$ -м концами создается связка, если: 1) расстояние между этими концами меньше определенного порогового  $d(i, j) < d_0$ ; 2) разность фаз лежит в определенных пределах  $\varphi_{min} < \varphi_i - \varphi_j < \varphi_{max}$ ; 3) отрезок, соединяющий эти два конца фрагментов изолинии, не пересекает линий скелета. Если между  $i$ -м и  $j$ -м концами создана связка, то значения полей  $count_i$  и  $count_j$  в массиве  $S_{ends}$  увеличиваются на единицу. Эту новую связку будем называть произошедшей от  $i$ -го и  $j$ -го концов.

**Шаг 3.** На данном шаге анализируется массив концов изолиний  $S_{ends}$ . Если в процес-

се анализа обнаруживается конец изолинии с номером  $i$  такой, от которого не произошла ни одна связка (поле  $count_i$  равно нулю), то это означает, что изображение недопустимо, и этот конец изолинии предъявляется оператору для коррекции. После коррекции алгоритм возвращается на выполнение шага 1.

**Шаг 4.** Проводится процедура сортировки связей. Она имеет следующий смысл. В массиве связей  $M_{conn}$  ищутся связки, которые не пересекаются с другими связками. Если такая связка находится, то она считается правильно распознанной и все другие связки, произошедшие от концов правильно распознанной связки, уничтожаются. При уничтожении связки между  $i$ -м и  $j$ -м концами значения полей  $count_i$  и  $count_j$  уменьшаются на единицу. В результате могут появиться новые связки, не пересекающиеся с другими. Процедура заканчивается, когда при очередном просмотре массива связей не была обнаружена ни одна новая не пересекающаяся с другими связка.

**Шаг 5.** На данном шаге снова анализируется список концов изолиний  $S_{ends}$ . При этом недопустимость или неоднозначная прослеживаемость скелетного массива может проявиться двумя способами. Первый заключается в том, что находится  $i$ -й конец изолинии, от которого не произошла ни одна связка (поле  $count_i$  равно нулю). Это означает, что изображение недопустимо. В этом случае конец изолинии предъявляется оператору для коррекции. После коррекции алгоритм возвращается на шаг 1. Второй способ: в процессе анализа обнаружен  $i$ -й конец изолинии, от которого произошла более чем одна связка (поле  $count_i$  больше единицы). Это означает, что изображение не является однозначно прослеживаемым, и данный конец изолинии также предъявляется оператору для коррекции. После коррекции выполнение алгоритма повторяется с шага 1. Если же у каждого конца изолинии имеется только одна производная связка ( $count_i = 1, i = 1 - N, \dots$ , где  $N$  — длина массива  $S_{ends}$ ), то это означает, что скелетный массив фрагментов изолиний является однозначно прослеживаемым. В этом случае связки вносятся в скелетный массив и прослеживание заканчивается.

**2.4. Ввод цифровых значений, соответствующих изолиниям.** После прослеживания изолиний необходимо каждой из них присвоить соответствующее числовое значение. Для этого оператор выполняет следующие действия.

1. Указывает маркером очередную изолинию. Выбранная изолиния отмечается на дисплее особым цветом.

2. В появившемся диалоговом окне вводит соответствующее числовое значение.

Для облегчения контроля выполнения этой процедуры при высвечивании изолиний программа использует три различных цвета. Изолинии, которым еще не присвоены числовые значения, высвечиваются одним цветом, выбранная изолиния — другим, а те изолинии, которым числовое значение уже присвоено — третьим. Благодаря этому оператор не пропустит ни одной изолинии и не будет присваивать значение одной и той же изолинии дважды.

**Формирование выходных результатов.** Кодирование карты завершается формированием файла, содержащего геометрическую (координаты концов отрезков изолиний) и цифровую информации об изолиниях. При этом геометрическая информация приводится к новой системе координат, начало которой находится в левом верхнем углу рамки, а направления осей соответствуют направлениям верхней и левой сторон рамки. Необходимые для такого преобразования данные были получены на этапе выделения рамки (см. раздел 2.2).

**Характеристика программного комплекса.** Описанная технология реализована в виде комплекса программ, работающих под управлением среды MS Windows. Комплекс имеет дружественный пользовательский интерфейс и включает в себя средства для удобного просмотра и редактирования растровых и векторных изображений.

Экспериментальная проверка алгоритма на реальных изображениях доказала, что в большинстве сложных ситуаций он надежно выполняет автоматическое прослеживание изолиний. Так, например, информационно насыщенный участок карты на рис. 3 прослеживается полностью автоматически и правильно. К достоинствам алгоритма можно отнести и то, что по ходу прослеживания выявляются ошибки предыдущих шагов и оператор может их исправить.

**ПОСТРОЕНИЕ ЦИФРОВОЙ МОДЕЛИ КАРТЫ.** Пусть некоторая регулярная сетка (например прямоугольная) покрывает всю область карты, изображение которой в результате описанных процедур находится в памяти компьютера. Необходимо по значениям изолиний и координатам точек, в которых изоли-

нии пересекают стороны сети, определить значение изображенной функции в ее узлах. Если выбрать некоторое скользящее окно, перемещать его по области карты, а в рамках окна каждый раз строить кубический сплайн  $Z(x, y)$ , то задача получает решение. Теория и техника построения кубических сплайнов двух переменных известна [10], но она довольно громоздка. Удовлетворительные результаты дает такая облегченная методика. Вдоль всех линий сети, параллельных координатной оси  $X$ , последовательно строится кубический сплайн  $Z_i(x)$ ,  $i = 1, n$ , где  $n$  — количество линий, и по его значениям в узлах сети  $(x, y)$ , через которые проходит данная линия, определяются значения функции  $Z_i(x, y)$ . Для построения сплайна удобно пользоваться программой Райнша [11], которая гарантировано дает надежные результаты. Затем та же операция выполняется для линий сети, параллельных оси  $Y$ , т. е. строятся сплайны  $Z_j(y)$ ,  $j = 1, m$ , и по ним определяются необходимые значения  $Z_j(x, y)$ . Искомые величины находятся как среднее из двух найденных для каждой точки узла, а именно:

$$Z(x, y) = \frac{1}{2} [Z_i(x, y) + Z_j(x, y)].$$

Как показала практика, значения угловых точек карты необходимо закреплять. По ним вначале определяются значения поля по сторонам рамки, а затем последние с повышенным весом используются для построения сплайнов  $Z_i(x)$ ,  $i = 1, n$ , и  $Z_j(y)$ ,  $j = 1, m$ . Таким образом, рамка карты служит "каркасом", на который "натягиваются" все сплайны.

**РЕЗУЛЬТАТЫ ОПРОБОВАНИЯ МЕТОДИК НА МОДЕЛЬНЫХ ПРИМЕРАХ.** Чтобы судить об эффективности работы методик, т. е. скорости ввода в компьютер изображения карты и точности построения по введенной информации их цифровой модели, выполним следующие модельные испытания.

**ТЕСТ 1.** Узлам квадратной сети  $5 \times 5$ , где шаг сети равен 3 км, присвоены некоторые значения глубин в км (рис. 4), отображающие достаточно быстро меняющийся рельеф некоторой поверхности (которая, как очевидно, представляется набором квадратных пластин). По значениям глубин в узлах сети вручную (очень тщательно и квалифицированно) строится карта (см. рис. 4), которая служит исходным материалом для последующих опробований. Их выполнено три: 1) по изолиниям карты рис. 4 вручную восстанавливаются значения поля в узлах сети и сравниваются с исход-

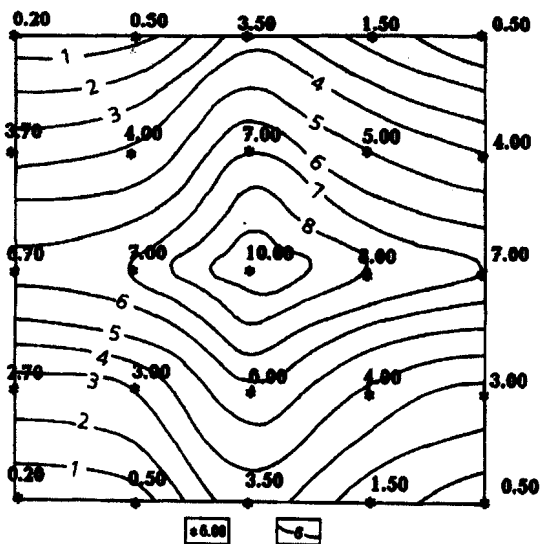


Рис. 4. Тест 1. Карта рельефа поверхности, построенная по значениям глубин, заданных в узлах сетки. Карта используется как исходная для опробования разработанных программных комплексов. 1 — 6,00 — значение глубины поверхности, км; 2 — изолиния и ее значение, км.

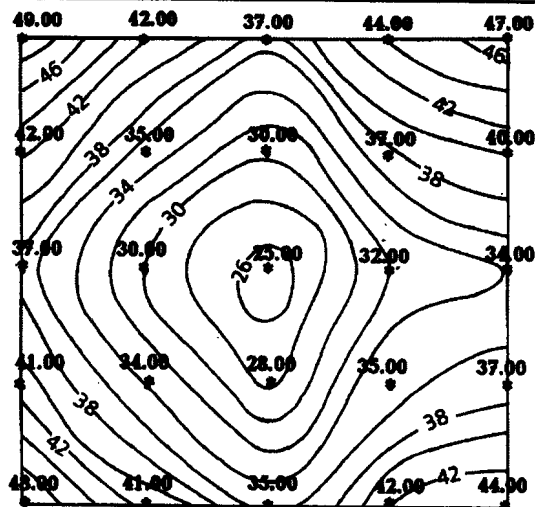
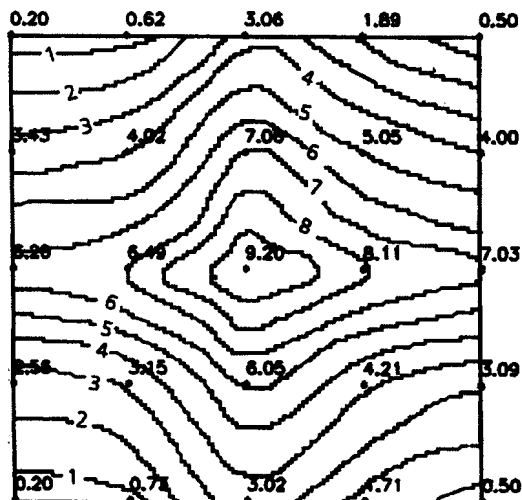
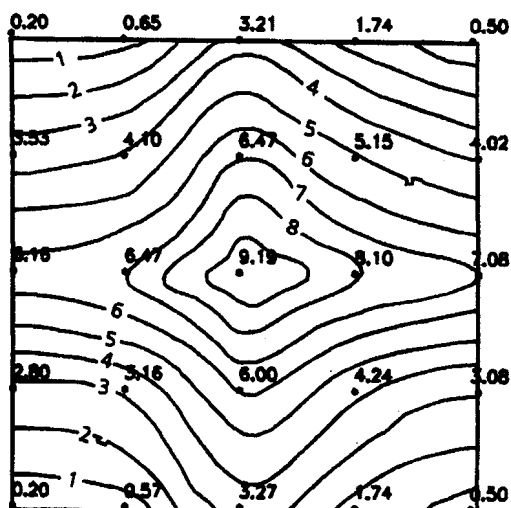


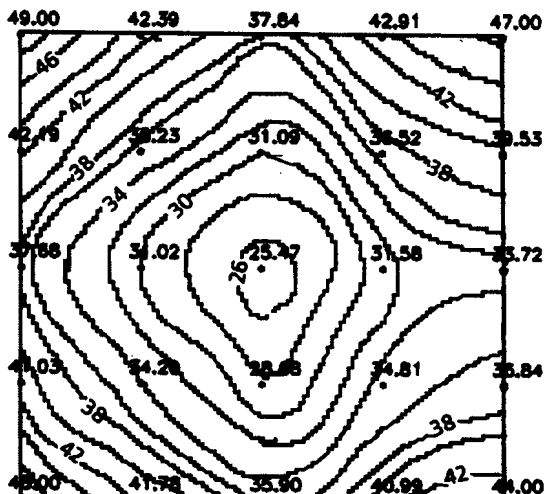
Рис. 6. Тест 2. Карта того же назначения, что карта рис. 4, но для иного рельефа поверхности. Обозначения те же, что и на рис. 4.



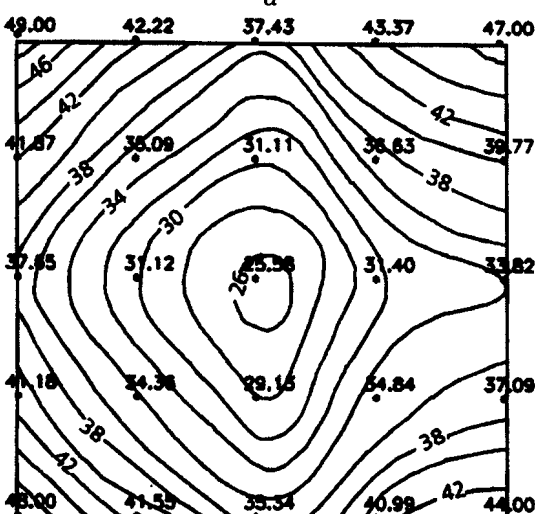
а



б



а



б

Рис. 5. Тест 1. Карты и их цифровые модели, полученные при вводе карты рис. 4 в компьютер с помощью дигитайзера(а) и сканера(б). Обозначения те же, что и на рис. 4.

Рис. 7. Тест 2. Карта и ее цифровые модели, полученные при вводе карты рис. 6 в компьютер с помощью дигитайзера(а) и сканера(б). Обозначения те же, что и на рис. 4.

Таблица сопоставления эффективности методик ввода в компьютер информационно простых карт с помощью дигитайзера и сканера

Время и точность выполнения операций	Время ввода карты в компьютер		Среднеквадратическая погрешность цифровой модели		
	с помощью дигитайзера, мин	с помощью сканера, мин	при ручном построении, км	при вводе карты с помощью дигитайзера, км	при вводе карты с помощью сканера, км
Тест 1	17	12	0.196	0.285	0.280
Тест 2	16	10	0.184	0.592	0.543

ными; 2) карта рис. 4 с помощью дигитайзера вводится в компьютер, строится ее цифровая модель и оценивается ее точность путем сравнения с данными рис. 4; 3) делается то же, что в пункте 2), но в компьютер карта вводится с помощью сканера. На рис. 5 приведены карты и восстановленные значения поля в узлах сети при вводе исходной карты в компьютер с помощью дигитайзера (см. рис. 5, а) и сканера (см. рис. 5, б). В таблице приведены данные о времени ввода карт и среднеквадратичной погрешности их цифровой модели, а также данные о скорости и точности выполненных операций.

**ТЕСТ 2.** Выполнены те же операции, что в тесте 1, но с еще более сложным рельефом поверхности. Карта рис. 6 — исходная. Карта рис. 7 — результат ввода в компьютер карты рис. 6 и построения ее цифровых моделей.

Из рассмотрения таблицы следует, что погрешности числовых моделей карт примерно одинаковые при их вводе в компьютер с помощью дигитайзера и сканера. Скорость ввода карт с помощью сканера выше. Очевидно также, что точность формальных методов построения цифровых моделей уступает ручным (при их тщательном и квалифицированном выполнении). Однако последние занимают слишком много времени и поэтому не могут быть конкурентноспособными на практике.

Необходимо также обратить внимание на следующее важное обстоятельство. Тесты 1 и 2 просты по информативной нагрузке и только при этом условии возможно сопоставление эффективности методик ввода в компьютер изображений карт с помощью дигитайзе-

ра и сканера. Если карты информационно насыщены (как, например, карта рис. 1), то ввести их в компьютер с помощью дигитайзера практически невозможно из-за неудовлетворительно низкой точности вводимой информации. В этом случае работа со сканером — самый быстрый, точный и надежный способ решения задачи. Следовательно, в практических условиях использование сканера имеет безусловные преимущества. Они также выше по сравнению с ручными способами.

**ЗАКЛЮЧЕНИЕ.** 1. Разработаны методики, алгоритмы и программные комплексы автоматизированного ввода в компьютер изображений карт с помощью дигитайзера и сканера. Программные комплексы работают под управлением MS Windows.

2. Выполнено опробование разработанных комплексов на двух тестовых примерах, которые информативно просты. Показано, что при указанном условии ввод карт в компьютер с помощью сканера выполняется быстрее, чем с помощью дигитайзера. Точность построения цифровых моделей карт практически одинакова при использовании как дигитайзера, так и сканера.

3. Для работы с информационно насыщенными картами, которые обычны для геофизической практики, использование сканера имеет безусловные преимущества.

4. Использование разработанных комплексов может иметь широкую область применения при проведении различных видов обработки и интерпретации геофизической информации, представленной в виде карт.

Список литературы

1. Старостенко В.И., Бас Р.Г., Бутаков Г.С., Дядюра В.А. Автоматизированная система оперативной обработки данных гравиметрии и магнитометрии. — Киев: Наук. думка, 1972. — 164 с.
2. Ablameyko S., Beregov B., Kryuchkov A. Computer-Added Cartographical System for Map Digitizing // Proceedings of International conference on documents analysis and recognition (ICDAR'93). — Tsukuba: Japan. — 1993. — P. 115—118.
3. Мацелло В.В., Шлезингер М.И. Синтаксический анализ изображений в процессе построчного сканирования // Изв. АН СССР. Техническая кибернетика. — 1982. — № 3. — С. 173—179.
4. Шлезингер М.И. Математические средства обработки изображений. — Киев: Наук. думка, 1989. — 198 с.
5. Мацелло В.В. Алгоритм удаления помех при обработке графических изображений // Автоматика. — 1986. — № 6. — С. 86—89.
6. Александрова С.Л. Быстродействующий алгоритм логической фильтрации бинарных изображений // Теоретические и прикладные вопросы распознавания изображений. — Киев: Ин-т кибернетики АНУ, 1991. — С. 25—35.
7. Kijko V.M., Schlesinger M.I. Width-independent fast skeletonization algorithm for binary pictures // International Journal of Imaging Systems and Technology. — 1991. — V.3. — P. 222—226.
8. Кийко В.М. Алгоритмы кусочно-линейной аппроксимации дискретизованных кривых // Методы и средства информатики речи. — Киев: Ин-т кибернетики АНУ, 1991. — С. 35—43.
9. Беллман Р. Дрейфус С. Прикладные задачи динамического программирования. — М.: Наука, 1965. — 455 с.
10. Cayley A. On contour and slope lines // Philosophical Magazine. — 1859. — V. 18. — P. 264—268.
11. Завьялов Ю.С., Квасов Б.И., Мирошниченко В.Л. Методы сплайн-функций. — М.: Наука, 1980. — 352 с.
12. Reinsch C.H. Smoothing by spline functions // Numerische Mathematic. — 1967. — V. 10. — P. 177—183.