УДК 681.3

ПОСТРОЕНИЕ ИЗОЛИНИЙ С АВТОМАТИЧЕСКИМ МАСШТАБИРОВАНИЕМ

А. В. Переберин[1]

В работе рассматривается применение B-сплайновых вейвлетов в задаче построения, сглаживания и отображения линий уровня (изолиний). Основанные на B-сплайновых вейвлетах методы обработки кривых обеспечивают эффективное масштабирование объектов, что позволяет осуществлять их отображение на области практически любого размера при относительно небольших временны́х затратах на пересчет в требуемом масштабе.

Ключевые слова: линии уровня, сплайновые кривые, B-сплайны, многомасштабный анализ, вейвлеты.

1. Введение. Сама по себе задача построения линий уровня совсем не нова, она возникала и возникает в самых различных областях — научной визуализации, геологии, картографии и т.д.

В определенной степени эту задачу можно считать решенной, как только на основе входных данных (как правило, это прямоугольная таблица значений исследуемого параметра) найдена последовательность точек, через которую нужно провести кривую — эта кривая и будет линией уровня. Задача построения кривой практически любой гладкости по заданной последовательности точек (сплайновой кривой) также неоднократно решалась множеством различных способов.

Однако на практике возникают дополнительные проблемы. Исходные данные, по которым должна быть построена кривая, могут содержать различные искажения. Например, в геологии в качестве исходных данных могут выступать результаты реальных замеров некоторого параметра, сделанные в "полевых" условиях. Построенные по таким данным линии будут искажены высокочастотным шумом, который желательно подавить.

Построение сплайновой кривой, как правило, состоит из вычисления коэффициентов полинома нужной степени на каждом фрагменте кривой и последующего вычисления этого полинома для заданных значений параметра — такой метод не всегда может оказаться эффективным.

В современных операционных системах с оконным интерфейсом размер окна, в котором осуществляется вывод, может быть изменен пользователем одним движением "мыши". Хотелось бы, чтобы изображение в окне могло бы автоматически перерисовываться с учетом изменения размера, причем время перерисовки должно быть как можно меньше.

Предлагаемый способ представления линий уровня в виде кубических B-сплайновых кривых [3, 7] дает возможность применить к ним так называемые B-сплайновые вейвлеты [4, 8] — гибкий инструмент, позволяющий, во-первых, сгладить кривую, а во вторых, эффективно отобразить кривую с любым разрешением, заданным пользователем. При этом алгоритм отображения настолько прост, что, после определенных доработок, может быть реализован аппаратно.

Дальнейшее изложение строится следующим образом. В п. 2 кратко описывается достаточно распространенный алгоритм построения последовательности управляющих точек линии уровня. В п. 3 конкретизируется задача обработки и отображения линии, обсуждается построение кубической B-сплайновой кривой по найденной последовательности и применение B-сплайновых вейвлетов для сглаживания, масштабирования и отображения кривой. В п. 4 демонстрируются результаты работы реализованного метода.

2. Построение последовательности управляющих точек. Опишем достаточно известный алгоритм построения последовательности управляющих точек линии уровня, т.е. точек, по которым впоследствии будет проведена кривая, являющаяся, по сути, искомой линией.

По условию задачи, исходными данными является прямоугольная таблица размера $M_x \times M_y$ значений некоторого параметра z. Всегда можно считать, что это таблица значений в точках с целыми координатами некоторой функции $z(x, y)$, определенной на прямоугольной области $\Pi = [0, M_x - 1] \times [0, M_y - 1]$, то есть

$$z(i, j) = z_{i,j}, \quad i = 0, \ldots, M_x - 1, \quad j = 0, \ldots, M_y - 1.$$

[1] Институт прикладной математики им. М. В. Келдыша РАН, Миусская пл., 4, 125047, Москва; e-mail: avpereb@newmail.ru

Множество точек $(i,j),\ i=0,\ldots,M_x-1,\ j=0,\ldots,M_y-1$, определяет на П *сетку* $\mathcal{K}$ (эти точки в таком случае будут именоваться *узлами* сетки). *Ребрами* сетки $\mathcal{K}$ назовем соединяющие узлы отрезки единичной длины. Ребра направлены вдоль оси OX или OY, но не по диагонали. Обозначаются ребра координатами пары узлов, которые они соединяют, например $(i,j)-(i,j+1)$ или $(i,j)-(i+1,j)$.

Пусть теперь требуется построить на П линию уровня некоторого значения Z функции $z(x,y)$.

Для каждого ребра сетки значения в узлах, которые ребро соединяет, сравниваются с Z. Если значение в одном из узлов оказалось больше Z, а в другом — меньше, то это значит, что линия уровня пересекает это ребро в некоторой точке. Если же оба значения больше (или меньше) Z, то линия уровня через это ребро не проходит.

Возможны еще две ситуации: значение одного из узлов ребра совпадает с Z и значения обоих узлов совпадают с Z. Вторая ситуация весьма маловероятна, но если она возникает, то можно, например, незначительно изменить значение в одном из узлов. При этом внесенная ошибка может быть сколь угодно малой, но формально равенство нарушится.

Если же равенство достигается только в одном из узлов, то эту ситуацию можно разрешить следующим образом. Введем понятия начала и конца ребра. *Началом* ребра будем считать тот узел (из двух возможных), сумма координат которого меньше, а *концом* ребра — оставшийся узел. Теперь можно договориться считать, что линия пересекает ребро, если совпадает с Z значение в начале ребра, и не пересекает, если с Z совпадает значение в конце ребра.

Допустим, что значение в некотором узле с координатами (i,j) равно Z. В этом узле сходятся четыре ребра: $(i-1,j)-(i,j)$, $(i,j-1)-(i,j)$, $(i,j)-(i+1,j)$ и $(i,j)-(i,j+1)$. Согласно сказанному выше, линия уровня не пересечет первые два ребра и пересечет вторые два ребра.

Если установлено, что линия пересекает ребро, то требуется найти координаты точки пересечения.

Проще всего в качестве такой точки брать середину ребра. Но это может оказаться достаточно грубым приближением. Кроме того, если одно и то же ребро пересекают линии разных уровней, то все они будут проходить через одну и ту же точку, что даст нежелательный эффект при отображении. Предлагаемый подход — линейно проинтерполировать функцию $z(x,y)$ вдоль ребра и найти точку, в которой интерполированная функция примет значение Z.

Таким образом, найдены все ребра, которые пересекаются изолинией (будем называть такое ребро *ребром с пересечением*), и для каждого из этих ребер найдены координаты точки пересечения.

Следующим этапом является соединение точек в последовательности. Перед тем как описать соответствующий алгоритм, заметим следующее.

Любые четыре узла вида (i,j), $(i+1,j)$, $(i,j+1)$ и $(i+1,j+1)$ являются вершинами квадрата, стороны которого образуют четыре ребра, соединяющие эти узлы. Каждое некраевое ребро является общей стороной двух таких квадратов, а каждое краевое ребро является стороной только одного квадрата.

Теперь опишем алгоритм построения последовательностей.

Шаг 0. Все ребра с пересечениями объявляются *непомеченными*, все ребра без пересечений — *помеченными*.

Шаг 1. Ищется любое непомеченное ребро (заметим, что любое непомеченное ребро — это заведомо ребро с пересечением). Оно помечается и объявляется *текущим*. Соответствующая точка пересечения инициализирует последовательность. Кроме того, ребро запоминается как условное начало линии. Любой квадрат, стороной которого является ребро, объявляется *текущим*, т.е. квадратом, в котором будет производиться поиск следующей точки. Если ребро оказалось краевым, то текущий квадрат определяется, естественно, единственным образом.

Шаг 2. Поиск следующей точки. В текущем квадрате ищутся стороны, которые являются непомеченными ребрами. Если поиск пересечений был выполнен правильно, то непомеченных ребер всегда должно быть либо одно, либо три. Исключение составляет случай, когда одно из помеченных ребер (кроме текущего) оказалось условным началом линии. Этот случай будет рассмотрен ниже.

Если не помечено только одно ребро квадрата, то соответствующая ему точка пересечения добавляется в последовательность, ребро помечается и становится текущим. (Это соответствует тому, что только одна линия пересекает квадрат.) Если непомеченных ребер три (это значит, что квадрат пересекают две линии), то в качестве следующей точки можно выбрать точку, соответствующую любому ребру, *не противолежащему* последнему помеченному ребру (в противном случае вторая линия должна будет пересечь первую, что не допускается). Если точка выбрана, то она добавляется в последовательность, соответствующее ребро помечается и становится текущим.

Если же одно из нетекущих ребер квадрата оказалось началом линии, то непомеченных ребер в квадрате либо может не быть ни одного, либо — два. В первом случае процесс построения последовательности

останавливается и последовательность объявляется *замкнутой*, т.к. ее конец совпал с началом. Во втором случае, при условии правильно выполненного поиска пересечений, ребро (начало линии) никогда не будет противолежащим текущему (всегда будет иметь с ним общий узел). Поэтому можно либо продолжить процесс, выбрав следующую точку на непротиволежащем непомеченном ребре, пометив его и сделав текущим, либо остановить процесс, объявив последовательность замкнутой.

Шаг 3. Текущее ребро может оказаться либо краевым, либо нет. Если ребро не краевое, то текущим объявляется квадрат, стороной которого является ребро, и отличный от того, который только что был текущим. После этого выполняется шаг 2.

Если ребро краевое, то последовательность объявляется *незамкнутой*. Текущим снова становится ребро — условное начало линии. Если оно краевое, то процесс построения останавливается. В противном случае текущим квадратом становится квадрат, стороной которого является это ребро, но который не был выбран в первый раз. После этого выполняются шаги 2 (причем каждая новая точка добавляется не в конец, а в начало последовательности) и 3 то тех пор, пока текущее ребро не станет краевым. На этом процесс построения останавливается.

Шаг 4. Если остались непомеченные ребра, то снова выполняется шаг 1.

В результате выполнения данного алгоритма получается одна или несколько последовательностей, каждая из которых соответствует одной линии, замкнутой или незамкнутой (т.е. упирающейся в границы области).

3. Построение линии. Рассмотрим более подробно формулировку задачи обработки и отображения линий, а также построение кубических B-сплайновых кривых и применение B-сплайновых вейвлетов для сглаживания, масштабирования и отображения кривых.

3.1. Уточнение формулировки задачи. Построим линию по заданной последовательности точек $\mathbf{C} = \{\mathbf{c}_j\}_{j=0}^{N-1},\ N \in \mathbf{Z},\ N > 0$, где $\mathbf{c}_j \in \Pi,\ j = 0, \ldots, N-1$. Точки последовательности $\mathbf{C}$ будем далее называть *управляющими точками* или *узлами*, саму последовательность $\mathbf{C}$ — *управляющей последовательностью*.

Проанализируем задачу и уточним ее формулировку.

Простейший способ построения кривой — последовательное соединение точек отрезками прямой, т.е. построение ломаной. Достоинством этого способа является его простота. Однако не менее очевидны серьезные недостатки такого подхода. Интуитивно понятно, что линия уровня должна быть по возможности "плавной". Достаточно неформальному понятию "плавности" в определенной степени соответствует математическое свойство гладкости кривой хотя бы первого порядка. Ломаная таким свойством не обладает (она лишь *кусочно*-гладкая). Можно допустить, что при определенных условиях отображенная ломаная будет выглядеть достаточно "плавной". Для этого, в частности, желательно, чтобы любые три подряд идущие точки $\mathbf{c}_{j-1}$, $\mathbf{c}_j$ и $\mathbf{c}_{j+1}$ последовательности лежали бы "почти на одной прямой", т.е. сумма длин отрезков $\overline{\mathbf{c}_{j-1}\,\mathbf{c}_j}$ и $\overline{\mathbf{c}_j\,\mathbf{c}_{j+1}}$ была бы близка к длине отрезка $\overline{\mathbf{c}_{j-1}\,\mathbf{c}_{j+1}}$. При определенном разрешении такая ломаная может выглядеть "плавной". Однако стоит только в несколько раз увеличить выводимый объект, как его кусочно-линейная "природа" станет очевидной. Таким образом, даже при самом удачном расположении узлов в управляющей последовательности, простое их соединение отрезками дает *плохо масштабируемый* результат.

Более предпочтительный вариант — построить по точкам управляющей последовательности кривую второго или третьего порядка гладкости, т.е. *сплайновую* кривую. Являясь более гладкой, чем ломаная, она значительно лучше масштабируется. Однако и с отображением гладких сплайновых кривых возникает ряд проблем. Выше было отмечено, что при удачном расположении управляющих точек даже ломаная может выглядеть достаточно плавной кривой. Но возможна и прямо противоположная ситуация, когда при неудачном расположении точек даже гладкая кривая, построенная по ним, будет выглядеть совсем не плавной. Такая ситуация особенно вероятна в случае "плохих", т.е. искаженных входных данных, о чем уже говорилось в п. 1. К этому также добавляется погрешность вычисления координат управляющих точек. Таким образом, реально найденные управляющие точки наверняка окажутся смещенными относительно управляющих точек, которые должна была иметь гипотетическая гладкая линия. Построенная по таким точкам линия будет искажена высокочастотным шумом, причем этот шум будет так или иначе проявляться вне зависимости от того, какой гладкости кривая будет построена по управляющей последовательности.

Получается, что перед тем, как вывести кривую, может понадобиться так преобразовать управляющую последовательность, чтобы подавить высокочастотный шум. Такая процедура носит название *сглаживания* кривой.

Таким образом, задача построения линии распадается на три подзадачи.

1. Сглаживание кривой. Модификация последовательности управляющих точек с целью подавления шумов и искажений.

2. Масштабирование кривой. Модификация последовательности управляющих точек с целью вывода кривой с разной степенью разрешения.

3. Отображение. Вывод на графическое устройство (или в графический файл) кривой, имея в качестве исходных данных последовательность управляющих точек.

Под модификацией последовательности точек подразумевается изменение координат точек последовательности, удаление точек, добавление новых.

3.2. B-сплайновые кривые и вейвлеты. B-сплайновые кривые [3, 7] вводятся как кривые, составленные из фрагментов — *элементарных B-сплайновых кривых*. Например, кубическая B-сплайновая кривая γ, заданная управляющей последовательностью $\mathbf{C}$, определяется как объединение элементарных кривых следующего вида:

$$\gamma_j(t) = \frac{(1-t)^3}{6}\,\mathbf{c}_{j-1} + \frac{3t^3-6t^2+4}{6}\,\mathbf{c}_j + \frac{-3t^3+3t^2+3t-1}{6}\,\mathbf{c}_{j+1} + \frac{t^3}{6}\,\mathbf{c}_{j+2}, \quad t\in[0,1], \quad j=0,\ldots,N-2. \tag{1}$$

Для построения кривых $\gamma_0(t)$ и $\gamma_{N-2}(t)$ требуется ввести два дополнительных узла. Для незамкнутых кривых они вводятся следующим образом:

$$\mathbf{c}_{-1} = (\mathbf{c}_0 - \mathbf{c}_1) + \mathbf{c}_0, \quad \mathbf{c}_N = (\mathbf{c}_{N-1} + \mathbf{c}_{N-2}) + \mathbf{c}_{N-1},$$

а для замкнутых кривых:

$$\mathbf{c}_{-1} = \mathbf{c}_{N-1}, \quad \mathbf{c}_N = \mathbf{c}_0.$$

B-сплайновая кривая является аппроксимирующей, а не интерполирующей, т.е. она не проходит через свои управляющие точки (если только они не лежат на одной прямой). Может показаться, что это не позволяет использовать эти кривые для построения линий уровня, если управляющие точки были найдены с помощью алгоритма, описанного в п. 2, поскольку эти точки должны принадлежать кривой. Тем не менее, именно B-сплайновые кривые были выбраны для построения линий уровня.

Выше было сделано предположение, что координаты точек были вычислены с ошибкой, т.е. они смещены относительно соответствующих точек искомой кривой. Величину и направление смещения определить невозможно, их можно считать случайными величинами. Эти случайные величины и порождают высокочастотный шум, искажающий кривую.

B-сплайновая кривая устроена таким образом, что каждая опорная точка влияет на форму кривой, т.е. "тянет" на себя соответствующий фрагмент кривой, однако фрагмент этой точки не достигает. Таким образом, B-сплайновая кривая в определенном смысле гасит разнос точек и, следовательно, изначально содержит в себе шумопонижающие (сглаживающие) свойства. Следовательно, применение B-сплайновых кривых в данном случае вполне оправдано.

Кроме того, следует заметить, что если управляющая последовательность оказалась "хорошей", т.е. любые три подряд идущие точки лежат "почти на одной прямой", что соответствует отсутствию (или очень низкому уровню) шума (см. п. 3.1), то кривая пройдет очень близко к этим точкам, и при построении кривой ошибка практически не будет заметна.

Однако основной аргумент в пользу B-сплайновых кривых — это возможность применить к ним вейвлет-анализ [1, 2, 5, 6], а именно, *B-сплайновые вейвлеты* [4, 8], с помощью которых предлагается решить все упомянутые выше задачи.

Обсудим иной вариант задания B-сплайновых кривых. Координаты узлов последовательности $\mathbf{C}$ рассматриваются как коэффициенты разложения векторнозначной функции $\gamma(t)$ по базису элементарных B-сплайнов, т.е.

$$\gamma(t) = \sum_{j=0}^{N-1} \mathbf{c}_j \mathcal{B}_j(t), \quad t\in[0,1]. \tag{2}$$

При правильном определении функций $\mathcal{B}_j(t)$ кривые, заданные формулами (1) и (2), совпадают, за исключением, возможно, фрагментов вблизи краевых узлов $\mathbf{c}_0$ и $\mathbf{c}_{N-1}$ (построение кривых вблизи краевых узлов будет рассмотрено ниже).

Обращаем внимание на то, что изменение любого узла приведет к изменению локального участка кривой, но не кривой в целом. Действительно, согласно формуле (1), любой узел $\mathbf{c}_j$ участвует в формировании только четырех фрагментов: $\gamma_{j-2}(t)$, $\gamma_{j-1}(t)$, $\gamma_j(t)$ и $\gamma_{j+1}(t)$, на формирование всех остальных

фрагментов кривой этот узел никак не влияет. Это же свойство сохраняется и в формуле (2), поскольку элементарные B-сплайновые функции $\mathcal{B}_j(t)$ имеют компактный носитель. Это свойство B-сплайновых кривых позволяет, в частности, утверждать, что любой способ построения кривой вблизи границ не изменит "внутреннюю" часть кривой.

Представление B-сплайновых кривых в форме (2) позволяет применить к этим объектам B-сплайновые вейвлеты. Заметим, что B-сплайновые вейвлеты порождаются многомасштабным анализом [5, 6], т.е. последовательностью вложенных подпространств специального вида, базисными функциями (скейлинг-функциями) которых являются элементарные B-сплайны.

Как известно, один шаг любого диадного вейвлет-преобразования проецирует элемент v пространства $V^{(i)}$ на два подпространства $V^{(i-1)}$ и $W^{(i-1)}$. Причем проекция на первое подпространство является "огрубленной" (низкочастотной) версией исходного элемента (это следует из определения и свойств многомасштабного анализа), а проекция на второе (детализирующее) подпространство содержит высокочастотную информацию, необходимую для восстановления исходного элемента по его проекции в $V^{(i-1)}$. Фактически преобразование применяется к коэффициентам разложения v по базисным функциям (скейлинг-функциям) $V^{(i)}$, а в результате получается набор коэффициентов разложения проекции элемента v на пространство $V^{(i-1)}$ по базисным функциям (скейлинг-функциям) этого пространства (низкочастотная составляющая) и набор коэффициентов разложения проекции элемента v на пространство $W^{(i-1)}$ по его базисным функциям (вейвлетам) (высокочастотная составляющая).

Таким образом, B-сплайновая кривая γ, определенная по формуле (2), является элементом одного из пространств $V^{(i)}$ многомасштабного анализа, а точки последовательности $\mathbf{C}$ суть коэффициенты ее разложения по базису скейлинг-функций (элементарных B-сплайнов) в этом пространстве.

Применим один шаг B-сплайнового вейвлет-преобразования к координатам узлов и рассмотрим полученные в результате коэффициенты низкочастотной составляющей. Как было замечено выше, эти коэффициенты суть коэффициенты разложения проекции исходной кривой на $V^{(i-1)}$. Но базисные функции $V^{(i-1)}$ также являются элементарными B-сплайнами (это опять-таки следует из определения многомасштабного анализа), то есть полученный набор коэффициентов есть набор узлов некоторой B-сплайновой кривой, а именно, проекции γ на $V^{(i-1)}$.

B-сплайновое преобразование, как и любое диадное вейвлет-преобразование, на каждом шаге каждой паре соседних элементов исходного сигнала (в данном случае сигналом является управляющая последовательность) ставит в соответствие один элемент низкочастотной части и один элемент высокочастотной части. Таким образом, каждый шаг прямого преобразования уменьшает количество узлов примерно вдвое (точное число узлов зависит от того, сколько узлов было в исходной последовательности и каким образом обрабатываются края кривой). При этом часть высокочастотной информации, естественно, теряется. Итак, с помощью B-сплайнового вейвлет-преобразования осуществляется *сглаживание* кривой.

Задачу построения B-сплайновой кривой можно было бы считать решенной, как только найдена соответствующая управляющая последовательность, при условии, что графическая библиотека и графическое устройство поддерживают работу с B-сплайновыми кривыми.

Более интересна ситуация, когда нет готового средства построения кривой и ее приходится строить поточечно. Простейший способ — построение в явном виде по формуле (1). Недостатки способа очевидны: для каждого звена кривой $\gamma_i(t)$ требуется вычислить коэффициенты соответствующего многочлена третьей степени и выбрать шаг изменения параметра t, затем для каждого текущего значения параметра вычислять значение этого многочлена.

Предлагается другой метод. Допустим, существует относительно недорогой способ изменения исходной последовательности узлов так, чтобы узлов становилось все больше, а построенная по ним кривая не менялась. С увеличением количества узлов управляющая ломаная приближается к самой кривой. Задача, таким образом, сводится к тому, чтобы, учитывая масштаб кривой и разрешение графического устройства, добавлять узлы до тех пор, пока построенная по ним ломаная сама по себе не окажется хорошей аппроксимацией искомой кривой. Преимущество такого метода еще и в том, что он применим не только для растровых, но и для векторных устройств, которые, в конечном счете, любой объект представляют в виде последовательности отрезков.

Операция по соответствующему изменению последовательности узлов носит название "вставки узла" и ее описание можно найти в соответствующей литературе [7]. В данном же случае интересно то, что́ представляет из себя вставка узла с точки зрения B-сплайновых вейвлетов.

Увеличение количества узлов без изменения кривой соответствует тому, что ту же самую кривую "переразлагают" по базисным функциям более высокого разрешения (частоты), при этом никакой новой информации не добавляется. Иными словами, кривую, представленную как элемент $V^{(i)}$, нужно пред-

ставить в пространстве $V^{(i+1)}$. Но для этого достаточно выполнить один шаг обратного B-сплайнового вейвлет-преобразования с высокочастотной составляющей, равной нулю.

Каждый шаг обратного преобразования увеличивает количество узлов в последовательности приблизительно вдвое (точное количество зависит от того, сколько точек было в исходной цепочке, и от того, как решено обрабатывать края кривой). При этом приблизительно вдвое сокращается расстояние между двумя соседними узлами. Таким образом, можно достаточно легко оценить максимально возможный размер отрезка ломаной. Например, при отображении на обычный экран монитора достаточно выполнять преобразование до тех пор, пока максимально возможная длина отрезка ломаной не будет соответствовать 4–5 пикселам. После этого можно просто вывести на экран полученную управляющую ломаную.

Может, однако, возникнуть и обратная ситуация, когда узлов изначально очень много, расстояние между ними мало и при отображении возможно наложение узлов друг на друга. Такое может случиться, если кривую требуется вывести с очень низким разрешением (например, в окне малого размера). В этом случае имеет смысл выполнять прямое преобразование (то есть то же, что делалось для сглаживания) до тех пор, пока отрезки не увеличатся до тех же 4–5 пикселов.

Таким образом, один и тот же инструмент — B-сплайновое вейвлет-преобразование — оказался применим для решения всех трех подзадач: сглаживания, масштабирования и отображения.

Теперь перейдем к более формальному описанию алгоритма.

3.3. Реализация вейвлет-преобразования. Как известно, в общем случае биортогональное одномерное диадное вейвлет-преобразование полностью определяется четырьмя *фильтрами*: низкочастотным анализа $\tilde{\mathbf{h}} = \{\tilde{h}_k\}_{k\in\mathbf{Z}}$, высокочастотным анализа $\tilde{\mathbf{g}} = \{\tilde{g}_l\}_{l\in\mathbf{Z}}$, низкочастотным синтеза $\mathbf{h} = \{h_m\}_{m\in\mathbf{Z}}$ и высокочастотным синтеза $\mathbf{g} = \{g_n\}_{n\in\mathbf{Z}}$. Используемые на практике фильтры содержат конечное число элементов, однако их всегда можно доопределить нулевыми элементами на всем множестве целых индексов.

Если известны коэффициенты всех фильтров, то преобразование выполняется следующим образом. Пусть дан некоторый дискретный сигнал $\mathbf{s}^{(i)} = \{s_j^{(i)}\}_{j\in\mathbf{Z}}$, Тогда один шаг прямого вейвлет-преобразования выделяет из этого сигнала низкочастотную составляющую $\mathbf{s}^{(i-1)}$ и высокочастотную составляющую $\mathbf{d}^{(i-1)}$ по формулам:

$$s_j^{(i-1)} = \sum_{k=-\infty}^{+\infty} \tilde{h}_k\, s_{2j+k}^{(i)}, \quad j \in \mathbf{Z}, \tag{3}$$

$$d_j^{(i-1)} = \sum_{l=-\infty}^{+\infty} \tilde{g}_l\, s_{2j+l}^{(i)}, \quad j \in \mathbf{Z}. \tag{4}$$

Шаг обратного преобразования (восстановление сигнала $\mathbf{s}^{(i)}$) определяется формулой

$$s_j^{(i)} = \sum_{m=-\infty}^{\infty} h_{j-2m}\, s_m^{(i-1)} + \sum_{n=-\infty}^{\infty} g_{j-2n}\, d_n^{(i-1)}, \quad j \in \mathbf{Z}. \tag{5}$$

Если элементы $\mathbf{s}^{(i)}$ суть коэффициенты разложения проекции некоторого объекта на пространство $V^{(i)}$ по базисным функциям этого пространства, то $\mathbf{s}^{(i-1)}$ содержит коэффициенты разложения проекции того же объекта на подпространство $V^{(i-1)}$ меньшего уровня разрешения по его базисным функциям, а $\mathbf{d}^{(i-1)}$ — коэффициенты разложения проекции объекта на детализирующее подпространство $W^{(i-1)}$. Формулы (3), (4) и (5) являются известными формулами *быстрого вейвлет-преобразования* (прямого и обратного). Очевидно, что если все фильтры имеют конечное число ненулевых элементов, то все суммы в указанных формулах также конечны.

Отметим, что элементами сигналов $\mathbf{s}^{(i)},\ i \in \mathbf{Z}$, и, следовательно, $\mathbf{d}^{(i)},\ i \in \mathbf{Z}$, могут быть не только скалярные значения, но и вектора (например, координаты точек на плоскости или в пространстве). Формулы быстрого вейвлет-преобразования содержат только линейные операции сложения и умножения на число, которые для векторов выполняются покоординатно. Следовательно, применение формул к векторнозначным сигналам эквивалентно их применению по отдельности для каждой координаты векторов-элементов.

Кубическим B-сплайновым вейвлетам соответствуют следующие коэффициенты фильтров:

$$\begin{aligned}
&\tilde{h}_0 = \frac{5}{4}, \quad \tilde{h}_{-1} = \tilde{h}_1 = \frac{5}{32}, \quad \tilde{h}_{-2} = \tilde{h}_2 = -\frac{3}{8}, \quad \tilde{h}_{-3} = \tilde{h}_3 = \frac{3}{32},\\
&\tilde{g}_0 = \frac{3}{8}, \quad \tilde{g}_{-1} = \tilde{g}_1 = -\frac{1}{4}, \quad \tilde{g}_{-2} = \tilde{g}_2 = \frac{1}{16};\\
&h_0 = \frac{3}{4}, \quad h_{-1} = h_1 = \frac{1}{2}, \quad h_{-2} = h_2 = \frac{1}{8},\\
&g_0 = \frac{5}{2}, \quad g_{-1} = g_1 = -\frac{5}{16}, \quad g_{-2} = g_2 = -\frac{3}{4}, \quad g_{-3} = g_3 = -\frac{3}{16}.
\end{aligned} \tag{6}$$

Теперь вернемся к задаче обработки и построения линий уровня.

В рассматриваемой задаче никак не используются высокочастотные компоненты. Действительно, при прямом преобразовании используется только низкочастотная составляющая сигнала, при обратном преобразовании никакой дополнительной информации не добавляется, т.е. высокочастотная составляющая тождественно равна нулю. Естественно, что по низкочастотному сигналу $\mathbf{s}^{(i-1)}$ и нулевой высокочастотной составляющей исходный сигнал $\mathbf{s}^{(i)}$ не может быть полностью восстановлен, вместо этого получается новый сигнал $\hat{\mathbf{s}}^{(i)}$. Таким образом, формула (4) при прямом преобразовании не используется, а для обратного преобразования достаточно только первого слагаемого формулы (5):

$$\hat{s}_j^{(i)} = \sum_{m=-\infty}^{\infty} h_{j-2m}\, s_m^{(i-1)}, \quad j \in \mathbf{Z}. \tag{7}$$

Также очевидно, что для реализации формул (3) и (7) требуется знать только два низкочастотных фильтра $\tilde{\mathbf{h}}$ и $\mathbf{h}$.

3.4. Преобразование управляющей последовательности. Исходным сигналом является управляющая последовательность $\mathbf{C}$, которая, в отличие от общего случая, описанного формулами (3), (4) и (5), всегда конечна. Подробно вопрос об обработке граничных узлов будет рассмотрен ниже.

Исходному сигналу поставим в соответствие некоторое произвольное значение уровня разрешения i, например $i = 0$. Таким образом, исходный сигнал получит обозначение $\mathbf{s}^{(0)}$. Элементами сигнала могут быть непосредственно точки последовательности $\mathbf{C}$, т.е.

$$s_j^{(0)} := \mathbf{c}_j, \quad j = 0, \ldots, N-1,$$

или какая-либо из координат этих точек, например координата x:

$$s_j^{(0)} := c_{x,j}, \quad j = 0, \ldots, N-1.$$

В силу сделанного выше замечания о векторных и скалярных элементах сигналов, второй вариант практически не отличается от первого, за тем исключением, что все последующие вычисления должны быть повторены и для y-координаты точек.

Далее для простоты изложения мы будем отождествлять сигналы $\mathbf{s}^{(i)}$ и $\hat{\mathbf{s}}^{(i)}$ с управляющими последовательностями.

Теперь рассмотрим правила обработки граничных узлов. При этом возникает две проблемы. Во-первых, доопределить исходный сигнал так, чтобы при выполнении преобразований не произошел выход за допустимые пределы индекса суммирования; во-вторых, выделить после выполнения преобразования те коэффициенты, которые сформируют новый сигнал.

Как было отмечено выше, управляющая последовательность $\mathbf{C}$ может задавать как замкнутую, так и незамкнутую кривую.

Начнем со случая замкнутой кривой.

Очевидно, что продолжение сигнала в этом случае должно быть периодическим. Действительно, если кривая замкнута, значит управляющая ломаная тоже замкнута. Следовательно, сигнал $\mathbf{s}^{(0)}$ можно доопределить так: $s_N^{(0)} := s_0^{(0)}$, $s_{N+1}^{(0)} := s_1^{(0)}$, … и $s_{-1}^{(0)} := s_{N-1}^{(0)}$, $s_{-2}^{(0)} := s_{N-2}^{(0)}$, …. Аналогичным образом доопределяется и любой сигнал $\mathbf{s}^{(i)}$, $i \in \mathbf{Z}$, когда к нему требуется применить преобразование.

Теперь рассмотрим вопрос о том, сколько элементов должно остаться у сигнала $\mathbf{s}^{(i-1)}$, если сигнал $\mathbf{s}^{(i)}$ насчитывает N_i элементов.

Один шаг любого диадного преобразования ставит в соответствие двум коэффициентам $s_{2j}^{(i)}$ и $s_{2j+1}^{(i)}$ один коэффициент $s_j^{(i-1)}$ (и один коэффициент $d_j^{(i-1)}$), что справедливо для любых целых i и j.

Допустим теперь, что число N_i четное. Нетрудно заметить, что в этом случае паре $s_0^{(i)}$ и $s_1^{(i)}$ будет соответствовать элемент $s_0^{(i-1)}$, паре $s_2^{(i)}$ и $s_3^{(i)}$ — элемент $s_1^{(i-1)}$ и т.д., а паре $s_{N_i-2}^{(i)}$ и $s_{N_i-1}^{(i)}$ будет соответствовать элемент $s_{N_i/2-1}^{(i-1)}$. Следующая пара коэффициентов в силу периодического продолжения совпадет с парой $s_0^{(i)}$ и $s_1^{(i)}$, и соответствующий им коэффициент $s_{N_i/2}^{(i-1)}$ совпадет с $s_0^{(i-1)}$. Таким образом, сигнал $\mathbf{s}^{(i-1)}$ в этом случае полностью определяется $N_i/2$ элементами с индексами от 0 до $N_i/2-1$.

Сложнее дело обстоит в случае, если N_i нечетно. Элемент $s_{\lfloor N_i/2\rfloor-1}^{(i-1)}$ соответствует паре $s_{N_i-3}^{(i)}$ и $s_{N_i-2}^{(i)}$, а следующий элемент $s_{\lfloor N_i/2\rfloor}^{(i-1)}$ соответствует паре $s_{N_i-1}^{(i)}$ и $s_0^{(i)}$, т.е. совпадения не происходит. С точки зрения сплайновой кривой такая ситуация означает, что две управляющие точки, определяемые элементам $s_0^{(i-1)}$ и $s_{\lfloor N_i/2\rfloor}^{(i-1)}$, соответствуют двум несовпадающим, но пересекающимся участкам кривой.

Простейший способ разрешить эту ситуацию — просто не включать один из конфликтующих элементов (например, с индексом $\lfloor N_i/2\rfloor$) в формируемый сигнал. Этот вариант удобен тем, что не требует отдельно рассматривать случаи четного и нечетного количества элементов. Для любого N_i будет справедливо соотношение $N_{i-1}=\lfloor N_i/2\rfloor$, и сигнал $\mathbf{s}^{(i-1)}$ будет состоять из элементов $s_0^{(i-1)}$, $s_1^{(i-1)}$, …, $s_{N_{i-1}}^{(i-1)}$.

Неплохой результат дает другой вариант. Вместо сигнала $\mathbf{s}^{(i-1)}$ формируется модифицированный сигнал $\bar{\mathbf{s}}^{(i-1)}$ такой же длины: $N_{i-1}=\lfloor N_i/2\rfloor$. Все элементы, кроме первого (с индексом 0), совпадают с элементами сигнала $\mathbf{s}^{(i-1)}$, а элемент $\bar{s}_0^{(i-1)}$ является средним арифметическим двух конфликтующих элементов, т.е.

$$\bar{s}_0^{(i-1)}=\frac{1}{2}\left(s_0^{(i-1)}+s_{N_{i-1}}^{(i-1)}\right).$$

Заметим, что формально этот вариант также подходит для случая четного N_i. Действительно, в этом случае $s_0^{(i-1)}=s_{N_{i-1}}^{(i-1)}$, следовательно $\bar{s}_0^{(i-1)}=s_0^{(i-1)}$ и $\bar{\mathbf{s}}^{(i-1)}$ просто совпадает с $\mathbf{s}^{(i-1)}$.

Очевидно, что при обратном преобразовании проблем с четными и нечетными длинами сигналов не возникает. Длина сигнала на каждом шаге удваивается, то есть будет заведомо четной. Это, в частности, означает, что длина сигнала $\hat{\mathbf{s}}^{(i)}$ может оказаться меньше, чем длина $\mathbf{s}^{(i)}$ до преобразования. Такая ситуация вполне допустима, учитывая то, что полное восстановление сигнала путем обратного преобразования в данной задаче не требуется.

Теперь перейдем к рассмотрению незамкнутой кривой. Строго говоря, это особый объект, который отличается от замкнутых кривых. Предполагается, что незамкнутая B-сплайновая кривая должна начинаться в первой управляющей точке и заканчиваться в последней (напоминаем, что в общем случае управляющие точки кривой не принадлежат). Дополнительно требуется, чтобы в начальной и конечной точках отрезки управляющей ломаной, соединяющие первый узел со вторым и последний с предпоследним, были бы отрезками касательных к кривой.

Существует специальный вид B-сплайновых вейвлетов, с помощью которых обрабатываются и строятся незамкнутые кривые [4, 8]. Эти кривые обладают двумя существенными недостатками. Во-первых, возникающие при их обработке системы не являются ни однородными, ни стационарными. Первое означает, что скейлинг-функции и вейвлеты (а следовательно, и соответствующие фильтры преобразований) могут изменяться в зависимости от индекса сдвига j; второе означает, что они также меняются в зависимости от индекса разрешения (масштаба) i. Это существенно усложняет вычисления, причем все усложнения вводятся только ради обработки нескольких узлов, близких к краевым, в то время как вдали от краев и функции, и фильтры ничем не отличаются от описанных выше. Вторым недостатком является строгое ограничение на количество узлов в последовательности. Для кубических B-сплайнов их должно быть 2^i+3 при любом разрешении $i\in\mathbf{Z}$.

Предлагается следующий способ преобразования незамкнутых кривых, который дает удовлетворительный результат, практически не усложняя вычислений.

Для доопределения сигнала используется константное продолжение, то есть для любого уровня разрешения $i\in\mathbf{Z}$ недостающие элементы сигнала определяются так: $s_j^{(i)}:=s_0^{(i)}$, $j<0$, и $s_j^{(i)}:=s_{N_i-1}^{(i)}$, $j\geqslant N_i$.

Элементы низкочастотной составляющей $\mathbf{s}^{(i-1)}$ определяются следующим образом: $s_0^{(i-1)}:=s_0^{(i)}$, $s_{\lfloor N_i/2\rfloor}^{(i-1)}:=s_{N_i-1}^{(i)}$, элементы с индексами от 1 до $\lfloor N_i/2\rfloor-1$ вычисляются по формуле (3). Следовательно, сигнал $\mathbf{s}^{(i-1)}$ содержит $N_{i-1}=\lfloor N_i/2\rfloor+1$ элементов, причем первый и последний элементы совпадают с первым и последним элементами исходного сигнала $\mathbf{s}^{(i)}$.

При обратном преобразовании коэффициенты определяются так: $\hat{s}_0^{(i)}:=s_0^{(i-1)}$, $\hat{s}_{2N_{i-1}}^{(i)}:=s_{N_{i-1}-1}^{(i-1)}$, а элементы с индексами от 1 до $2N_{i-1}-1$ вычисляются по формуле (3). Тем самым сигнал $\hat{\mathbf{s}}^{(i)}$ содержит

$2N_{i-1}+1$ элементов (всегда нечетное количество), причем первый и последний элементы не меняются.

Таким образом, показано, как с помощью прямого и обратного вейвлет-преобразований представить кривую с любым разрешением i (напомним, что для исходной последовательности было решено положить $i=0$). При этом на каждом шаге прямого преобразования часть высокочастотной информации теряется и количество управляющих точек уменьшается. На каждом шаге обратного преобразования количество управляющих точек увеличивается, а сама кривая при этом практически не меняется.

3.5. Сглаживание кривой. Как было отмечено выше, B-сплайновая кривая уже в определенной степени является сглаживающей по отношению к своей управляющей последовательности, однако на практике может потребоваться и более существенное сглаживание. Для этого выполняется фиксированное (заданное пользователем) количество шагов прямого преобразования $i_0 > 0$. Полученный сигнал $\mathbf{s}^{(-i_0)}$ содержит опорные точки сглаженной кривой.

Отметим следующее. B-сплайновые вейвлеты, которым соответствуют фильтры (6), относятся к классу так называемых *полуортогональных* вейвлетов [4, 8]. Этот класс обладает следующим свойством: для любого уровня разрешения $i \in \mathbf{Z}$ пространства $V^{(i)}$ и $W^{(i)}$ ортогональны друг другу (хотя базисы в каждом из этих пространств не обязаны быть ортогональными). Это означает, что $W^{(i)}$ является ортогональным дополнением $V^{(i)}$ до $V^{(i+1)}$ и один шаг прямого вейвлет-преобразования с фильтром $\tilde{\mathbf{h}}$, коэффициенты которого приведены в (6), реализует ортогональное проектирование элемента из $V^{(i+1)}$ на $V^{(i)}$. Как известно, ортогональная проекция на пространство является наилучшим приближением элемента в этом пространстве. Это свойство полезно, в частности, при сжатии информации, когда требуется уменьшить объем данных, стараясь при этом минимизировать потери информации. В задаче же сглаживания, наоборот, требуется избавиться от высокочастотной информации, поэтому имеет смысл применить какой-нибудь другой фильтр, с лу́чшими сглаживающими свойствами, но, тем не менее, не вносящий существенных искажений в низкочастотную составляющую сигнала.

Для этих целей можно использовать, например, фильтр со следующими коэффициентами (этот фильтр также соответствует определенному виду вейвлетов):

$$\tilde{h}_0 = \tilde{h}_1 = 0{,}7031, \quad \tilde{h}_{-1} = \tilde{h}_2 = -0{,}1094, \quad \tilde{h}_{-2} = \tilde{h}_3 = -0{,}1406, \quad \tilde{h}_{-3} = \tilde{h}_4 = 0{,}0469. \tag{8}$$

Этот фильтр (назовем его дополнительным) показал неплохие результаты при сглаживании сильно искаженных сигналов. В то же время, если уровень искажений не высок, то имеет смысл применять фильтр $\tilde{\mathbf{h}}$, коэффициенты которого приведены в (6) (будем называть его основным).

3.6. Масштабирование и отображение кривой. Рассмотрим процесс отображения кривой по имеющейся управляющей последовательности. Предлагается добавлять в управляющую последовательность узлы так, чтобы сама кривая не менялась, а управляющая ломаная приближалась к кривой. Такое добавление будет выполняться до того момента, когда вместо кривой можно будет отобразить управляющую ломаную. Именно на этом этапе кривая адаптируется к текущим параметрам вывода (предыдущие вычисления их никак не учитывали).

Пусть требуется отобразить кривую в окне размера $L_x \times L_y$ пикселов. Вспомним, что управляющая последовательность строилась по сетке размера $M_x \times M_y$. Если эту сетку отобразить на окно, то расстояние между двумя соседними узлами сетки не будет превышать $d = \max(M_x/L_x,\, M_y/L_y)$ пикселов[2] (это число можно округлить до ближайшего целого, но оно может оставаться и дробным). Очевидно, что размер в пикселах любого звена управляющей ломаной, построенной по этой сетке, также не будет превышать d. Один шаг прямого преобразования, примененного к управляющей последовательности, увеличивает это расстояние приблизительно вдвое, шаг обратного преобразования уменьшает это расстояние также приблизительно вдвое.

Таким образом, считая, что d соответствует уровню разрешения $i=0$, приблизительный пиксельный размер звена для каждого уровня $i \in \mathbf{Z}$ можно вычислить по формуле

$$d_i = 2^{-i} d, \quad i \in \mathbf{Z}.$$

В частности, сглаженной кривой, заданной последовательностью $\mathbf{s}^{(-i_0)}$, соответствует число $d_{-i_0} = 2^{i_0} d$.

Пусть теперь задано число d_{max}, определяющее максимально допустимый размер звена, при котором ломаная с нужным качеством визуально приближает кривую. Как правило, для растрового дисплея берется число d_{max}, равное 4–5 пикселам.

Теперь не составляет труда найти такой уровень разрешения $i = i_1$, что $d_{i_1} \leqslant d_{max}$, а $d_{i_1-1} > d_{max}$. Управляющая ломаная, построенная по сигналу (последовательности узлов) $\hat{\mathbf{s}}^{(i_1)}$, окажется той самой ломаной, которую можно отобразить вместо соответствующей кривой.

[2]Очевидно, что под расстоянием в данном случае понимается не математическая длина отрезка, а количество пикселов, необходимых для построения этого отрезка, без 1, т.е. максимум модуля разности координат пикселов-концов.

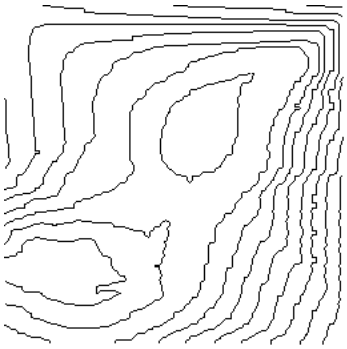

Рис. 1. Линии без сглаживания (искаженные данные)

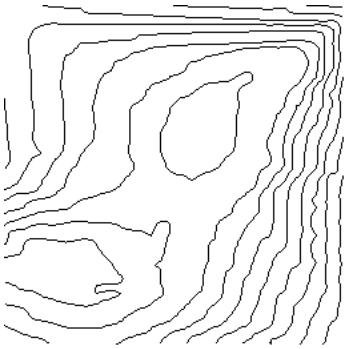

Рис. 2. Сглаживание (2 шага, основной фильтр)

Как правило, оказывается, что для сглаженной кривой, которой соответствует сигнал $\mathbf{s}^{(-i_0)}$, число $d_{-i_0} > d_{max}$, т.е. $-i_0 < i_1$. Следовательно, чтобы получить сигнал $\hat{\mathbf{s}}^{(i_1)}$, требуется применить к сигналу $\mathbf{s}^{(-i_0)}$ $i_0 + i_1$ шагов обратного преобразования.

Может, однако, оказаться, что $d_{-i_0} < d_{max}$ и $-i_0 > i_1$ (например, было сильно уменьшено окно, в котором требуется отобразить кривую). Очевидно, что в этом случае можно применить к $\mathbf{s}^{(-i_0)}$ $-(i_0 + i_1)$ шагов прямого преобразования. Однако в этом случае требуется не столько сглаживание сигнала, сколько уменьшение объема данных для его представления; следовательно, рекомендуется использовать только основной фильтр ортогонального проектирования, а не дополнительный сглаживающий.

Очевидно, что если $-i_0$ оказалось равным i_1, то дополнительных преобразований выполнять не нужно.

Отдельно можно упомянуть случай, когда $d_{max} = 1$. Это значит, что размер звена ломаной не превышает одного пиксела, то есть вместо ломаной достаточно отобразить все ее узлы, что в свою очередь будет соответствовать поточечному отображению кривой. Очевидно, что вычислительные затраты и объем памяти для хранения данных в этом случае вырастут, а скорость вывода заметно снизится по сравнению с отображением ломаной. Однако, во-первых, этот случай соответствует максимальному разрешению, с которым можно отобразить кривую при данных параметрах графического устройства. Во-вторых, поточечный вывод позволяет применить подавление лестничного эффекта при построении кривой, если такая функция не поддержана графическим устройством; правда, при этом скорость вывода снизится еще существеннее.

4. Реализация и результаты. Данный метод построения и отображения линий уровня был реализован в приложении, работающем в операционной системе MS Windows, без использования каких-либо дополнительных графических библиотек, кроме стандартных функций графического вывода, поддерживаемых этой системой (отобразить пиксел, определить цвет отображенного пиксела, отобразить отрезок).

Было реализовано построение управляющей последовательности, сглаживание, отображение линии с помощью управляющей ломаной, а также поточечное отображение с подавлением лестничного эффекта.

К приложению предъявлялись следующие требования: пользователь должен иметь возможность менять степень сглаживания; построенные линии уровня должны отображаться на экране в окне произвольного размера и при изменении размера окна пользователем линии должны перерисовываться со-

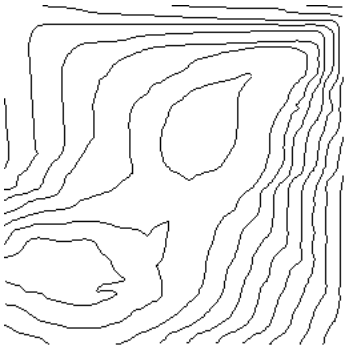

Рис. 3. Сглаживание (2 шага, дополнительный фильтр)

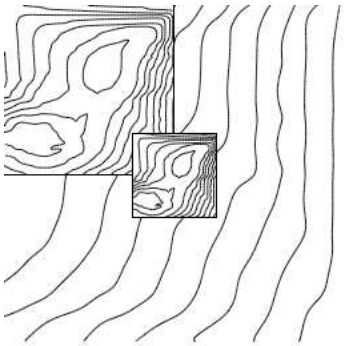

Рис. 4. Подавление лестничного эффекта и масштабирование

гласно новым размерам; кроме того, пользователь должен иметь возможность записать изображение в графический файл формата BMP (размер изображения при этом определяет сам пользователь).

На рис. 1 изображены линии, построенные по исходным данным (размер сетки 100×100) без дополнительной обработки. Для демонстрации возможностей сглаживания намеренно взяты данные, дающие сильно зашумленный результат. На рис. 2 и 3 показаны линии после двух шагов сглаживания основным и дополнительным фильтрами соответственно. Все линии на рис. 1–3 представлены своими управляющими ломаными (размер звена не превышает 4 пиксела). На рис. 4 изображены те же линии, что и на рис. 3, но в разных масштабах (как уменьшенные, так и фрагмент увеличенного изображения); линии построены поточечно, с подавлением лестничного эффекта.

СПИСОК ЛИТЕРАТУРЫ

1. *Новиков И.Я., Стечкин С.Б.* Основные конструкции всплесков // Фундаментальная и прикладная математика. 1997. **4**, № 3. 999–1028.
2. *Петухов А.П.* Введение в теорию базисов всплесков. СПб.: Изд-во СПбГТУ, 1999.
3. *Шикин Е.В., Боресков А.В.* Компьютерная графика. Динамика, реалистические изображения. М.: ДИАЛОГ-МИФИ, 1996.
4. *Chui C.K.* An Introduction to Wavelets. New York–London: Academic Press, 1992.
5. *Jawerth B., Sweldens W.* An overwiew of wavelet based on multiresolution analysis // SIAM Rev. 1994. **3**, N 3. 377–412.
6. *Mallat S.* A Wavelet Tour of Signal Processing. New York–London: Academic Press, 1998.
7. *Rogers D.* Introduction to NURBS: With Historical Perspective. San Francisco: Morgan Kaufmann, 2000.
8. *Stollnitz E.J., Derose T.D., Salesin D.H.* Wavelets for Computer Graphics. Theory and Applications. San Francisco: Morgan Kaufmann, 1996.

Поступила в редакцию
18.01.2001