

O. R. Musin, A. U. Ukhalov, H. Edelsbrunner, O. P. Yakimova, Fractal and Computational Geometry for Generalizing Cartographic Objects, *Model. Anal. Inform. Sist.*, 2012, Volume 19, Number 6, 152–160

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use
<http://www.mathnet.ru/eng/agreement>

Download details:

IP: 98.84.25.165

November 2, 2024, 07:29:40



УДК 517.51+514.17

Применение методов фрактальной и вычислительной геометрии для картографической генерализации линейных объектов

Мусин О. Р.* , Ухалов А. Ю.** , Эдельсбруннер Г.*** , Якимова О. П.**¹

* *The University of Texas at Brownsville, Texas, USA*

** *Ярославский государственный университет им. П. Г. Демидова,*

*** *Institute of Science and Technology, Klosterneuburg, Austria*

e-mail: polya@uniyar.ac.ru

получена 12 сентября 2012

Ключевые слова: картографическая генерализация, сегментация, кривизна, фрактальная размерность

Предлагается новый алгоритм генерализации линейных картографических объектов. Основным новшеством алгоритма является автоматическая сегментация – разбиение ломаной на участки с одинаковыми свойствами. Сегментация позволяет подобрать параметры сглаживания индивидуально для каждого участка, за счет чего существенно повышается качество результата.

Процесс генерализации играет важную роль в представлении карт и других геометрических данных. Генерализация – это обобщение изображаемых на карте объектов в соответствии с назначением карты, масштабом отображения и особенностями территории (см. [1]). В течение долгого времени генерализация считалась субъективным процессом, требующим участия квалифицированного специалиста. В связи с развитием вычислительной техники и средств автоматизации возникла необходимость формализации этого процесса. Особенно актуальной эта задача стала при распространении систем автоматической навигации и электронных карт, доступных через интернет. Невозможно хранить на сервере данные для отображения сразу во всех масштабах, которые могут быть затребованы пользователем. Естественным оказывается хранить данные, необходимые для представления в самом подробном масштабе, и осуществлять масштабирование на лету. Одной из трудностей, возникающих при автоматическом масштабировании, является избыточность данных. Попытка отобразить на экране размером в несколько сантиметров все данные приводит к тому, что результат становится неудобочитаем. Ломаная из нескольких тысяч (или десятков тысяч) звеньев, которая хорошо представляет гладкую кривую

¹Работа выполнена при поддержке гранта Правительства РФ по постановлению №220, договор № 11.G34.31.0053.

на большом экране, на маленьком выглядит как широкая полоса. Кроме того, передача лишних точек создает дополнительную нагрузку на сети передачи данных. Для отображения объекта на карте небольшого размера может быть достаточным сохранить лишь несколько десятков точек. Основная трудность при решении задачи состоит в выборе параметров упрощения. На участках, которые визуальнo воспринимаются как более «сложные» или «извилистые», необходимо сохранить больше точек, в то время как на более «гладких» участках можно удалить значительное количество точек без существенной потери информации.

В данной работе предлагается новый алгоритм для выполнения автоматической генерализации.

1. Алгоритм генерализации

Предлагаемый алгоритм состоит из следующих шагов:

1. Приведение ломаной к равнозвенному виду.
2. Сегментация – разбиение линии на сегменты, на которых кривизна ломаной приблизительно одинакова.
3. Вычисление фрактальной размерности каждого сегмента.
4. Упрощение сегментов с помощью алгоритма Дугласа–Пейкера [3]. Параметры упрощения выбираются автоматически, индивидуально для каждого сегмента, с учетом фрактальной размерности.
5. Сглаживание полученной ломаной с помощью В-сплайнов.

Основная идея предлагаемого алгоритма состоит в процедуре автоматической сегментации – выделения частей ломаной с одинаковыми свойствами.

2. Приведение ломаной к равнозвенному виду

Для вычислений в представленных ниже алгоритмах необходимо, чтобы исходная ломаная была равнозвенной, то есть расстояние между любыми двумя последовательно взятыми точками ломаной было одно и то же.

Входные данные алгоритма: исходная ломаная $L = (p_1, p_2, \dots, p_n)$.

Выходные данные: равнозвенная ломаная $L' = (p_1, p_2, \dots, p_{n'})$.

Опишем процедуру построения равнозвенной ломаной пошагово.

1. Вычислим величину шага δ . Обозначим через d_i расстояние между точками p_i и p_{i+1} .

$$d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}.$$

Среднее расстояние между точками $d_{in} = \frac{\sum_{i=1}^{n-1} d_i}{n-1}$. Отсюда $\delta = C d_{in}$. Здесь C – некоторая константа $0,2 \leq C \leq 1$. Ее введение необходимо из-за того, что при

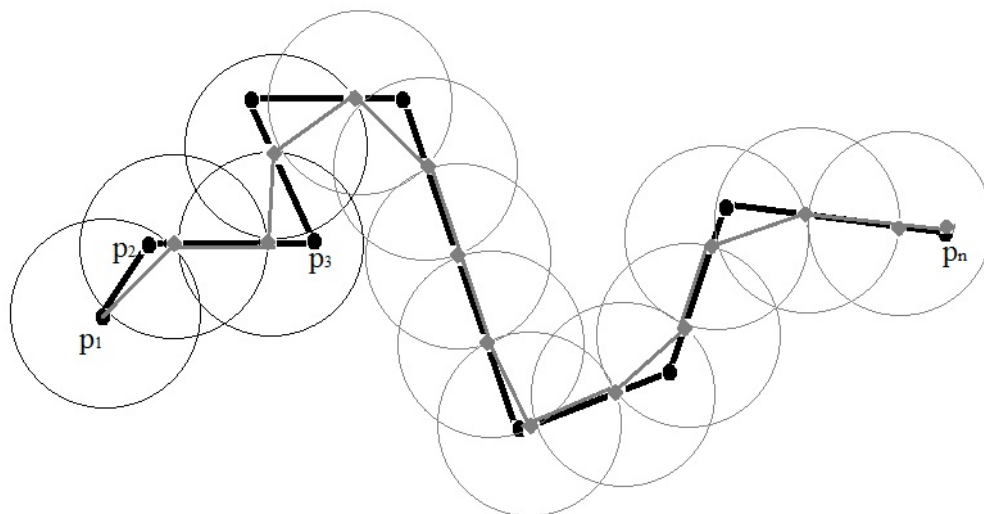


Рис. 1. Приведение ломаной к равнозвенному виду: черная линия – исходная ломаная, серая – полученный результат

съемке реального географического объекта точки ломаной расположены неравномерно и средний шаг не всегда отображает реальную плотность точек. По нашему предположению, $C = 0,5$.

2. Первую точку исходной ломаной p_1 поместим в результирующее множество точек L' .

3. Из последней точки результирующего множества проведем окружность радиуса δ и найдем точки пересечения окружности с исходной ломаной.

4. Если такие точки есть, то выберем из них ближайшую по ходу следования исходной ломаной и занесем в результирующий набор. Затем перейдем к шагу 3.

5. Если точек пересечения с исходной ломаной больше нет, то последняя точка исходной ломаной добавляется в результирующий набор и на этом работа процедуры заканчивается.

На рисунке 1 изображен пример построения равнозвенной ломаной.

Фактически переход от исходной ломаной к равнозвенной является предварительной обработкой исходных данных. Для упрощения обозначений будем равнозвенную ломаную обозначать так же, как исходную.

3. Сегментация

Входные данные алгоритма: равнозвенная ломаная $L = (P_1, \dots, P_n)$ ($P_i \in \mathbf{R}^2$), где $|P_{i+1} - P_i| = \rho, \forall i = 1, \dots, n-1$, N_{init} – число точек на сегменте в исходном разбиении на сегменты, N_p – минимальное число точек на сегменте, N_s – максимальное число сегментов, f – параметр, характеризующий вес числа экстремальных вершин.

Выходные данные: набор сегментов – равнозвенных ломаных, представляющих собой разбиение исходной ломаной.

Алгоритм сегментации

1. Начальное разделение ломаной линии на сегменты $L = \cup S_i$. Выбирается положительное N_{init} , и каждый начальный сегмент S_i состоит из N_{init} точек: $S_i = (P_{i,1}, \dots, P_{i,N_{init}})$.
2. Для каждой внутренней точки сегмента $P_{i,j} = (x_{i,j}, y_{i,j})$, ($1 < j < N_{init}$) вычисляется абсолютная величина угла поворота $C_{i,j}$ — дополнение угла $\alpha_{i,j}$ (угол при вершине $P_{i,j}$) до π , т.е.

$$C_{i,j} = \arccos \frac{(x_{i,j+1} - x_{i,j})(x_{i,j} - x_{i,j-1}) + (y_{i,j+1} - y_{i,j})(y_{i,j} - y_{i,j-1})}{\rho^2}.$$

3. Вычисляется полная кривизна сегмента

$$C_i = \sum_{1 < j < N_{init}} C_{i,j}.$$

Величина $C_i/(2\pi)$ называется *полной вариацией* сегмента.

4. Вычисляем число экстремальных вершин E_i — число локальных максимумов и локальных минимумов функции кривизны $\{C_{i,j}, 1 < j < N_{init}\}$, т.е. число вершин $P_{i,j}$ таких, что $C_{i,j} > C_{i,j-1}, C_{i,j} > C_{i,j+1}$ или $C_{i,j} < C_{i,j-1}, C_{i,j} < C_{i,j+1}$.
5. Вычисление интегральной характеристики сегмента $M_i = W_i + fE_i$, где f — некоторая положительная константа, параметр алгоритма.
6. Последовательно объединяем сегменты с минимальным различием в значениях характеристики M_i до тех пор, пока имеются сегменты, число точек в которых меньше, чем данное число N_p , и пока число сегментов больше, чем число N_s .

Результат работы алгоритма сегментации показан на рисунке 2.

4. Вычисление фрактальной размерности каждого сегмента

Определение величины фрактальной размерности каждого сегмента линейного объекта необходимо для его качественного упрощения при переходе к более мелкому масштабу.

Здесь под фрактальной размерностью понимается размерность Минковского d_M [5].

$$d_M(A) = - \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log \varepsilon},$$

где A — множество в R^2 , $N(\varepsilon)$ — количество клеток со стороной ε необходимых, чтобы покрыть множество A .

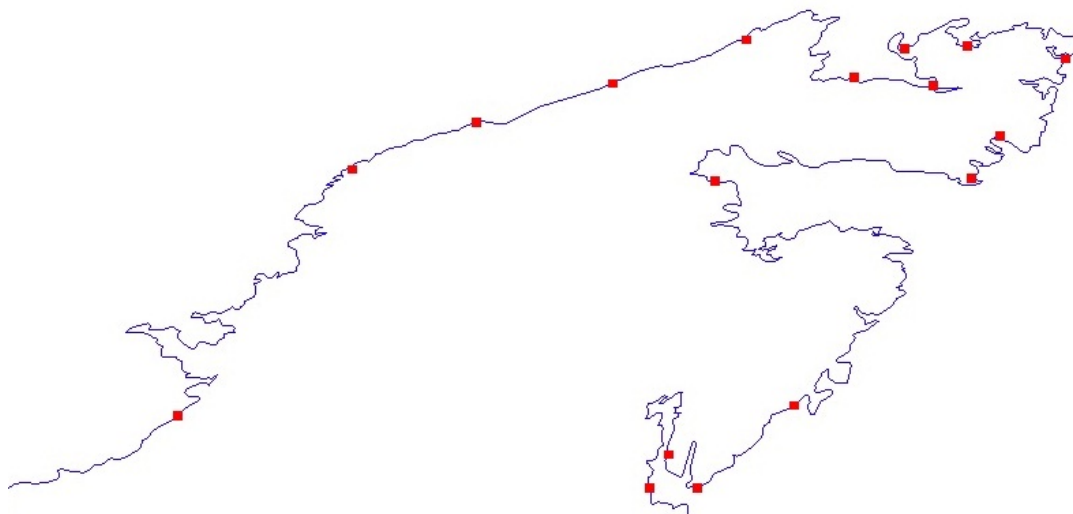


Рис. 2. Пример разделения береговой линии на сегменты

Для проведения компьютерных вычислений общепринятым является следующий способ: разобьем область, содержащую A (в нашем случае A – сегмент ломаной) на квадратные клетки со стороной ε , где ε принимает значения от δ (шаг ломаной) до $k * \delta$ (k – константа, причем $2 \leq k \leq 15$, в нашей программе $k = 8$). Затем считаем число клеток, необходимых для покрытия A в каждом случае. Из полученного набора точек (т.е. пар $(\varepsilon, N(\varepsilon))$) методом наименьших квадратов находим d_M .

5. Упрощение

Самым распространенным методом упрощения линейных данных считается алгоритм Дугласа–Пейкера [3]. Суть алгоритма состоит в том, чтобы по данной ломаной, аппроксимирующей кривую, построить ломаную с меньшим числом точек. Упрощенная ломаная состоит из подмножества точек исходной. У этого алгоритма всего один параметр – ширина коридора или полосы допуска h , которая зависит от масштаба.

Принцип работы алгоритма Дугласа–Пейкера представлен на рисунке 3. Начальная ломаная представляет собой упорядоченный набор точек, соединенных линиями на расстоянии $\delta > 0$. Она показана на рисунке 3а сплошной линией.

Входом алгоритма служат координаты всех точек между первой и последней. Первая и последняя точки сохраняются неизменными. На первом шаге алгоритм находит точку, наиболее удаленную от прямой, проведенной через первую и последнюю. На рисунке 3а это точка C . Если данная точка находится на расстоянии больше чем h , то алгоритм рекурсивно вызывает себя на наборах от начальной до данной и от данной до конечной точки, и данная точка сохраняется (рисунок 3б).

Если же данная точка находится на расстоянии, меньше чем h , то все точки,

которые ещё не были отмечены к сохранению, могут быть выброшены из набора и получившаяся прямая сглаживает кривую с точностью не ниже h . На рисунке 3в такими являются участки ломаной CF и FK . По окончании всех рекурсивных вызовов выходная ломаная строится только из тех точек, что были отмечены к сохранению (рисунок 3г).

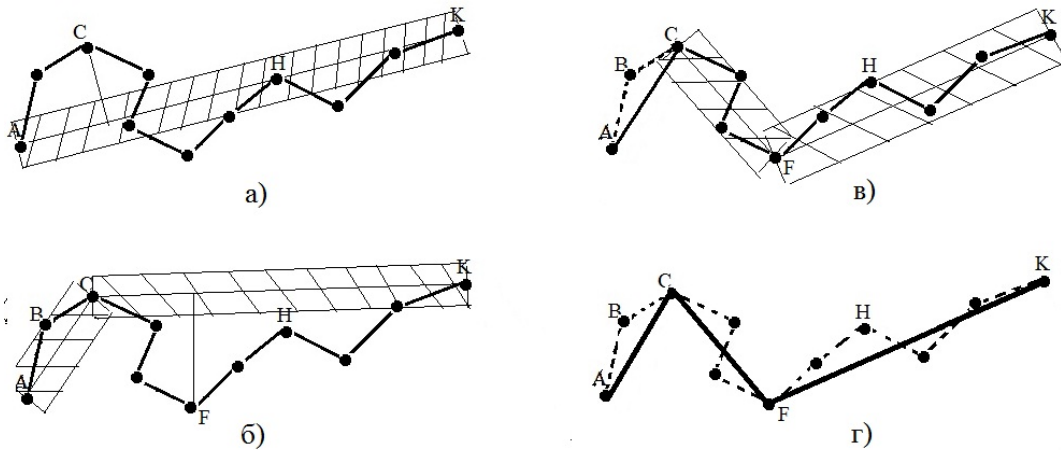


Рис. 3. Упрощение линейного объекта с помощью алгоритма Дугласа–Пейкера

В классическом варианте ширина коридора h зависит только от выходного масштаба карты m и равна величине, соответствующей 0,1 мм в масштабе m . 0,1 мм – это графическая точность карты, то есть печатный станок две линии, удаленные друг от друга на 0,1 мм, напечатает как две линии, а вот если между ними 0,05 мм, то линии уже сольются в силу ограничений разрешающей способности. При картографической генерализации сохраняются только те точки исходной ломаной, которые определяют основные изгибы линии. Чем меньше масштаб, тем меньше надо оставлять точек, иначе линия будет чрезмерно детализирована.

В нашем алгоритме мы предлагаем вычислять ширину коридора h для каждого сегмента отдельно, причем она зависит не только от масштаба m , но и от фрактальной размерности $d_M(S_i)$ сегмента S_i .

$$h(S_i, m) = c \cdot m^{2-d_M(S_i)}, \quad \text{где } c - \text{константа.}$$

Для более гладких участков ширина коридора будет больше, для сильно изрезанных – соответственно меньше, что позволит лучше отобразить характер линии.

6. Сглаживание сплайнами

После упрощения линейного объекта проводится его сглаживание. При упрощении уменьшается количество точек линии, при сглаживании уменьшается ее кривизна, то есть «угловатость» с геометрической точки зрения. С эстетической точки зрения сглаживание улучшает восприятие линейных элементов и читаемость карты.

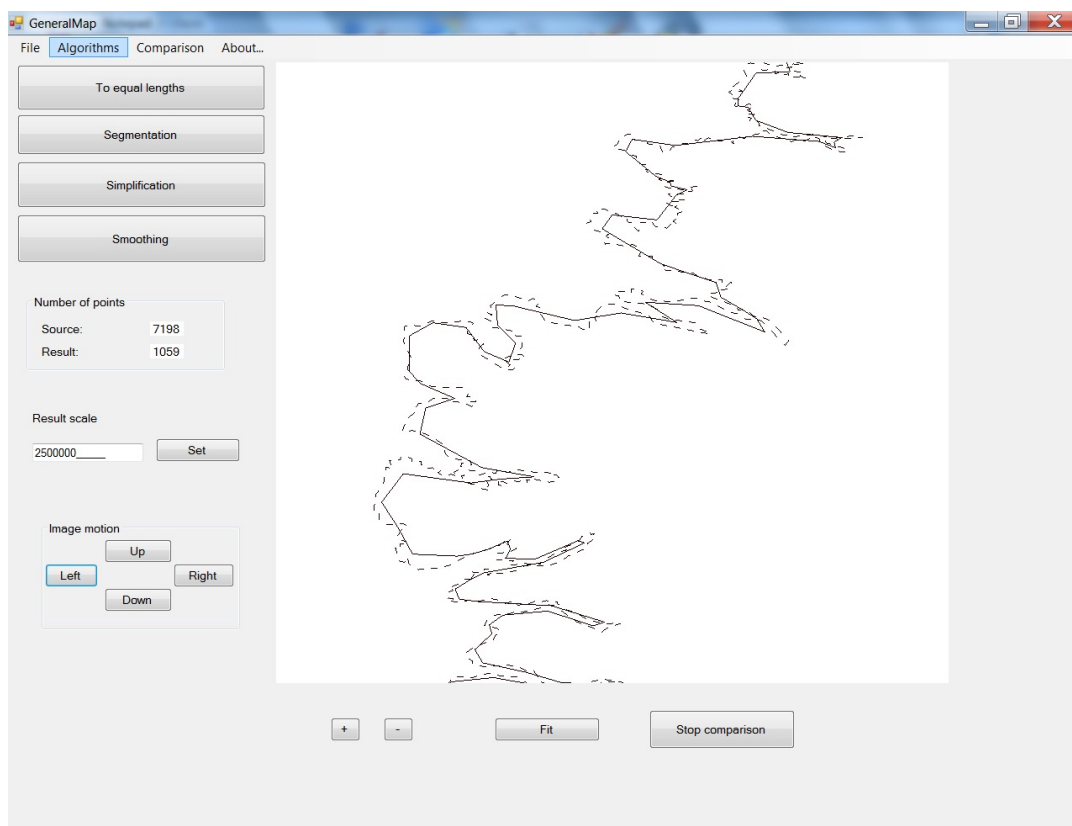


Рис. 4. Генерализация береговой линии Англии. Пунктирная линия – исходная ломаная, сплошная – результат, полученный с помощью предлагаемого алгоритма

В нашем алгоритме мы использовали метод сглаживания кривой с помощью кубических B -сплайнов (см., например, [1]). В этом случае вычисление координат вершин сглаженной ломаной (\bar{x}_i, \bar{y}_i) производится по формулам

$$\bar{x}_i = (x_{i-1} + 4x_i + x_{i+1})/6, \quad \bar{y}_i = (y_{i-1} + 4y_i + y_{i+1})/6.$$

Заметим, что линия, составленная из B -сплайнов, не будет проходить точно через заданные точки.

7. Программа GeneralMap

Авторами была разработана компьютерная программа GeneralMap, реализующая вышеописанный алгоритм картографической генерализации и позволяющая производить сравнение полученных результатов.

В качестве примера в данной работе мы рассматриваем генерализацию ломаной, представляющей собой береговую линию Англии. В исходном масштабе ломаная состоит из 7198 точек. При уменьшении масштаба от 1: 1500000 до 1 :2500000 наша программа оставляет 1059. При этом параметры сегментации выбираются автоматически. Результат работы программы приведен на рисунке 4.

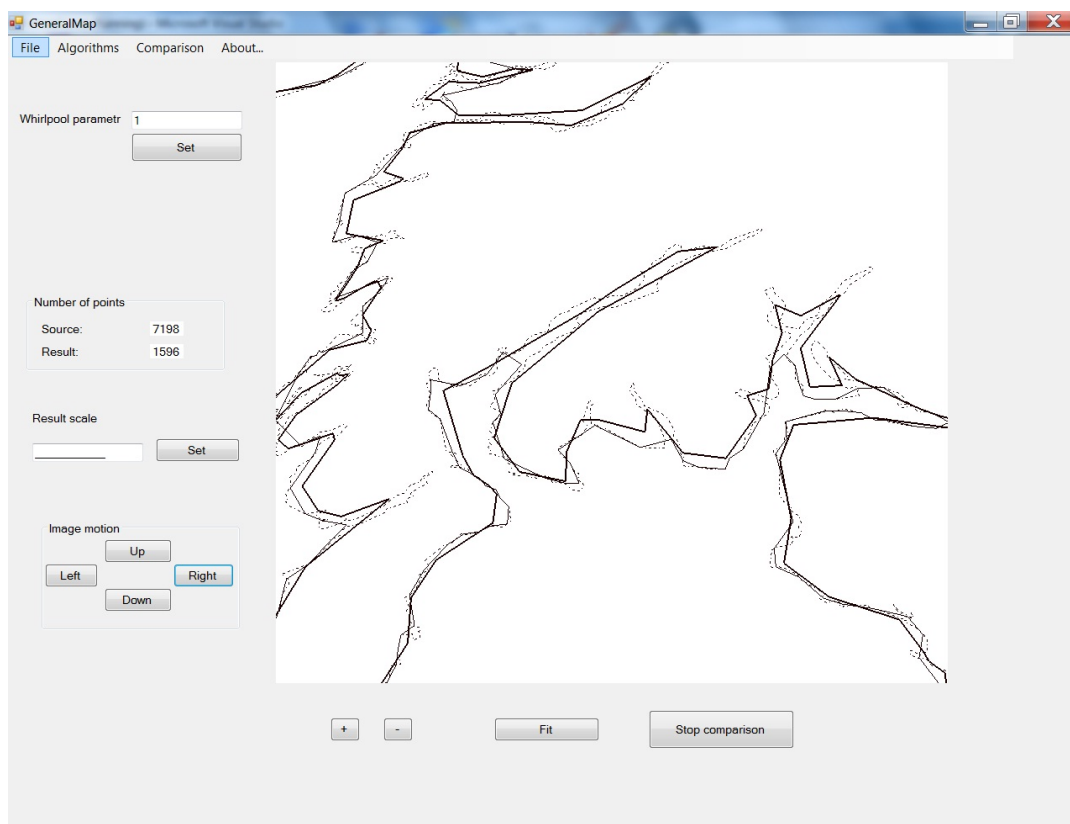


Рис. 5. Сравнение результатов генерализации береговой линии Англии: пунктирная линия – исходная ломаная, тонкая – результат вихревого алгоритма, жирная – алгоритма авторов

Сравнение с некоторыми существующими алгоритмами генерализации показывает, что предложенный в данной работе метод работает не хуже, а в ряде случаев – лучше, чем другие алгоритмы.

Е. Fritsch [2] предложил «вихревой» (whirlpool) алгоритм. Сравнение результатов работы алгоритмов приведено на рисунке 5. На тех же исходных данных вихревой алгоритм оставляет большее количество точек (1596), но при этом менее точно передает характер береговой линии.

В тестировании алгоритмов и в разработке программы принимали участие студенты – сотрудники лаборатории «Дискретная и вычислительная геометрия» имени Б.Н. Делоне – Алексеев В. В., Богаевская В. Г., Горохов А. А. и Преображенская М. М.

Список литературы

1. Берлянт А. М., Мусин О. Р., Собчук Т. В. Картографическая генерализация и теория фракталов. М., 1998.

2. Fritsch Emmanuel. Use of Whirlpool algorithm for ADBS data generalization // ADBS meeting. 1999.
3. Douglas D. H., Peucker T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature // Canadian Cartographer. 1973. Vol. 10, N 2. P. 112–122.
4. Bin Jiang, Xintao Liu, Tao Jia. Scaling of Geographic Space as a Universal Rule for Mapping or Cartographic Generalization. 2011. <http://arxiv.org/abs/1102.1561v1>
5. Кроновер Р.М. Фракталы и хаос в динамических системах. Основы теории. М.: Постмаркет, 2000.

Fractal and Computational Geometry for Generalizing Cartographic Objects

Musin O.R., Ukhalov A.U., Edelsbrunner H., Yakimova O.P.

Keywords: cartographic generalization, segmentation, curvature, fractal dimension

We present an algorithm for simplifying linear cartographic objects and results obtained with a computer program implementing this algorithm.

Сведения об авторах:

Мусин Олег Рустумович,

The University of Texas at Brownsville, Texas, USA

Ярославский государственный университет им. П.Г. Демидова,
научный сотрудник Международной лаборатории "Дискретная и вычислительная
геометрия" им. Б.Н. Делоне.

Ухалов Алексей Юрьевич,

Ярославский государственный университет им. П.Г. Демидова,
научный сотрудник Международной лаборатории "Дискретная и вычислительная
геометрия" им. Б.Н. Делоне.

Эдельсбруннер Герберт,

Австрийский Институт Науки и Технологий, Клостенбург, Австрия
(Institute of Science and Technology, Klosterneuburg, Austria);
Ярославский государственный университет им. П.Г. Демидова,
руководитель Международной лаборатории "Дискретная и вычислительная
геометрия" им. Б.Н. Делоне.

Якимова Ольга Павловна,

Ярославский государственный университет им. П.Г. Демидова,
научный сотрудник Международной лаборатории "Дискретная и вычислительная
геометрия" им. Б.Н. Делоне.