19. O m a c h i, S. A fast algorithm for a *k*-NN classifier based on branch and bound method and computational quantity estimation. http://citeseer.ist.psu.edu/611715.html.

Статья поступила в редакцию 5.07.2005

Валерий Михайлович Черненький родился в 1941 г., окончил МВТУ им. Н.Э. Баумана в 1964 г. Д-р техн. наук, профессор, заведующий кафедрой "Системы обработки информации и управления" МГТУ им. Н.Э. Баумана. Академик Международной академии информатизации. Автор 105 научных работ в области моделирования и системного анализа.

V.M. Chornenkiy (b. 1941) graduated from the Bauman Moscow Higher Technical School in 1964. D. Sc. (Eng.). professor, head of "Systems of Data Processing and Control" department of the Bauman Moscow State Technical University. Academician of the International Academy of Information Technology. Author of 105 publications in the field of modeling and system analysis.

Николай Валентинович Птицын родился в 1979 г., окончил МГТУ им. Н.Э. Баумана в 2002 г. Аспирант кафедры "Системы обработки информации и управления" МГТУ им. Н.Э. Баумана. Автор 8 научных работ в области системного анализа.

N.V. Ptitsyn (1979) graduated from the Bauman Moscow State Technical University in 2002. Post-graduate of "Systems of Data Processing and Control" department of the Bauman Moscow State Technical University. Author of 8 publications in the field of system analysis.

УДК 519.6

Н. В. П т и ц ы н

РАЗНОСТНЫЙ МЕТОД ИНТЕГРАЛЬНОГО ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЯ

Рассмотрен метод дискретного интегрального преобразования, основанный на вычислении разности между соседними суммами. Исследовано приложение метода к обработке изображения. Представлено сравнение с известными алгоритмами быстрой свертки.

Дискретные интегральные преобразования [1, 2] играют важную роль в задачах распознавания образов. С их помощью строятся фильтры для подавления шума, выделения признаков, сегментации и других операций, связанных с обработкой сигналов, таких как звук и изображение. Интегральные преобразования являются ресурсоемкими, особенно для многомерных сигналов и больших ядер. Настоящая статья представляет новый метод интегрального преобразования изображения, который в определенных случаях превосходит существующие методы с точки зрения точности, удобства реализации и быстродействия.

Для одномерного сигнала дискретное интегральное преобразование в общем случае есть сумма

$$s_n = \sum_{n'} f_{n'} g_{n,n'}, \quad 0 \le n < N, \quad 0 \le n' < N, \tag{1}$$

где f_n — исходный сигнал, содержащий N дискретных значений, s_n — результат преобразования и $g_{n,n'}$ — ядро преобразования.

Частным случаем интегрального преобразования является свертка, ядро которой может быть представлено как

$$g_{n,n'} = c_{n'-n} = c_k, \quad 0 \le k < K, \tag{2}$$

где $c_{n'-n}$ — константы, определяемые только разностью $k = (n' - n)$ и K — размер ядра (максимальная разность). Тогда выражение (1) примет вид

$$s_n = \sum_{n'} f_{n'} c_{n'-n}.$$

Число элементарных умножений и сложений, необходимое для получения всех значений s_n, равно NK.

Разностный метод. В основе предлагаемого метода лежит тот факт, что суммы $s_n, s_{n+1}, \ldots$ обычно вычисляются последовательно друг за другом. Разность между соседними суммами представим как

$$\Delta s_n = s_n - s_{n-1} = \sum_{n'} f_{n'} c_{(n'-n)} - \sum_{n'} f_{n'} c_{(n'-n-1)} =$$
$$= \sum_{n'} f_{n'} \left(c_{(n'-n)} - c_{(n'-n-1)} \right) = \sum_{n'} f_{n'} \Delta c_{(n'-n)},$$

где

$$\Delta c_{(n'-n)} = \Delta c_k = \begin{cases} -c_{0,} & k = 0, \\ c_{(k-1)} - c_k, & 0 < k \le K, \\ c_{(k-1)}, & k = K. \end{cases}$$

Метод с вычислением разностей эффективен в том случае, когда сложность (число операций) вычисления разности s_n меньше сложности получения всей суммы s_n. Это имеет место, когда разностная последовательность $(c_0 \ldots c_K)$ является сильно разреженной, то есть содержит достаточное количество нулевых элементов, позволяющих сократить вычисления.

Разностный подход ранее применялся для усреднения одномерных сигналов [2]. Свертка с прямоугольным импульсом *произвольной* длины K получается путем вычитания $f_{(n-K/2-1)}$ (в точке, покидающей окно) и добавления $f_{(n+K/2)}$ (в точке, присоединяющейся к окну):

$$s_n = s_{n-1} + \frac{1}{K} \left(-f_{(n-K/2-1)} + f_{(n+K/2)} \right).$$

Таким образом, необходимо $3N$ элементарных сложений в независимости от длины K.

Сравнение с известными методами. В работах [1, 3] рассматриваются два подхода непосредственной свертки (метод перекрытия с суммированием и метод перекрытия с накоплением) и три подхода быстрой свертки. Методы быстрой свертки различаются требуемым объемом вычислительных операций и памяти, а также степенью точности, связанной с ошибками округления.

Первый подход, основанный на быстром преобразовании Фурье (БПФ), приводит к существенному сокращению требуемого количества арифметических операций для $K > 32$ [4, 5]. Недостатки этого метода — значительные ошибки округления, большой объем памяти, требуемый для хранения комплексных экспоненциальных коэффициентов, и все еще значительный объем вычислений. В лучшем случае метод сокращает число операций до $2N \log_2 N + N$, что необходимо для прямого и обратного преобразований и перемножения коэффициентов.

Второй подход, использующий теоретико-числовые преобразования (ТЧП), является точным, так как служит для преобразования последовательностей в кольце целых чисел. Существенный недостаток, ограничивающий его применение в реальных системах, — зависимость между длиной последовательности N и требуемой длиной кодового слова, что приводит к длинным кодовым словам для больших N.

Третий подход основан на методах модульной арифметики в кольце полиномов, обеспечивающих высокие эффективность и точность вычислений. Недостаток этих методов заключается в сложности программирования вычислений, которая зависит от длины обрабатываемой последовательности.

Локальная свертка. Общим недостатком трех рассмотренных быстрых методов является сложность локального интегрирования “на месте”. Практические приложения, такие как распознавание образов с сегментированием, часто требуют локальную обработку сигнала в некоторой области S, часто заданной бинарной маской:

$$s_n = \sum_{n' \in S} f_{n'} c_{n'-n}.$$

В методах, основанных на преобразованиях (БПФ, ТЧП), происходит дополнение области до прямоугольной нулевыми или средними значениями. Это приводят к искажению исходного сигнала и существенным ошибкам на границах.

Преимущества метода свертки с вычислением разностей: высокая эффективность для свертки больших последовательностей с повторяющимися значениями; высокая точность для последовательностей це-

лых чисел; возможность локальной свертки без преобразований (таких как БПФ и ТЧП); минимальное число обращений к элементам последовательности и маски, задающей область свертки; простота программирования и отсутствие ограничений на длину последовательности (в отличие от методов, использующих кольцо целых чисел и модульную арифметику); возможность бóльшего сокращения операций за счет разделения ядра (для многомерных сигналов).

Метод имеет и недостатки, а именно, во-первых, он эффективен только тогда, когда разностная последовательность $(c_0 \dots c_K)$ разрежена, т.е. содержит существенные нулевые области. Если все $c_k \neq 0$, то требуется больше вычислительных операций, чем непосредственная свертка. Таким образом, этот метод имеет узкое применение. Во-вторых, "бегущая сумма" s_n будет накапливать ошибку, если в процессе вычисления s_n происходит округление. Поэтому метод не приемлем для вычислений с плавающей точкой.

Приложение к обработке изображения. Основным приложением метода свертки с вычислением разностей является обработка изображения. В задачах распознавания образов мы часто сталкиваемся с большими ядрами, состоящими из нулей и единиц. Пусть исходное изображение задано матрицей $|f_{m,n}|$, где координаты $0 \le m < M$, $0 \le n < N$; ядро задано матрицей $|c_{k,l}|$, где $0 \le k, l < K$, и результат свертки записывается в матрицу $|s_{m,n}|$. Двумерная дискретная свертка определяется выражением

$$s_{m,n} = \sum_{m'} \sum_{n'} f_{m',n'} c_{m'-m,\, n'-n}. \qquad (3)$$

Для реализации двумерной свертки методом разностного интегрирования необходимо: 1) определить способ сканирования (то есть путь окна по исходному изображению) таким образом, чтобы текущая сумма вычислялась из предыдущей за минимальное число операций; простейший способ — сканирование зигзагом (рис. 1); 2) вычислить предварительно разностную матрицу $|c_{k,l}|$ для каждого из возможных направлений движения; в случае сканирования зигзагом таких матриц будет три (влево, вниз и вправо); 3) определить ненулевые элементы в разностных матрицах и записать их в индексный массив. Особенно выгодно предварительно рассчитать готовое смещение относительно начала (центра) матрицы ядра; 4) реализовать цикл сканирования, в котором будут использоваться индексы (смещения) разностной матрицы в зависимости от направления движения.

На рис. 2 приведена разностная матрица при движении налево диска радиуса $R = 4{,}5$. Всего ядро имеет 57 ненулевых точек, в то время как разностная матрица только 18. Таким образом, выигрыш в числе операций составляет $(57 - 18)/57 = 64\,\%$.

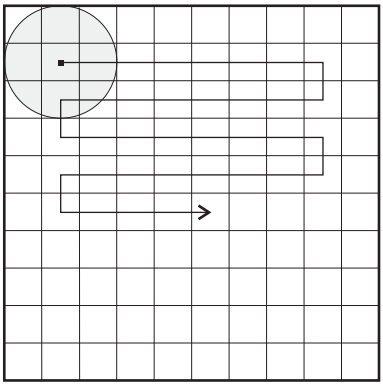

Рис. 1. Разностная матрица ядра “диск” при движении окна направо. Пунктирными квадратами обозначены ненулевые разности

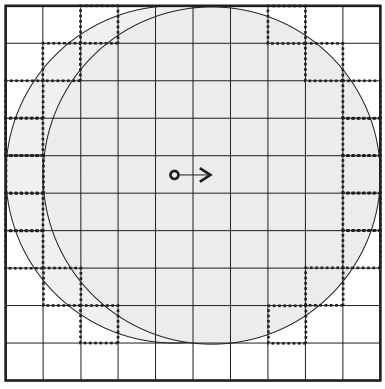

Рис. 2. Сканирование изображения окном для вычисления свертки разностями

Несмотря на некоторую громоздкость такого подхода, алгоритм имеет серьезные преимущества при реализации локальной свертки. Он минимизирует число обращений к исходному изображению и число проверок принадлежности точки к области свертки.

В случае локальной свертки в окне число точек в окне может меняться. По этой причине необходимо нормализировать сумму фактическим числом точек $w_{m,n}$, то есть рассматривать отношение $s_{m,n}/w_{m,n}$. Значение $w_{m,n}$ определяется из своего предыдущего значения, аналогично $s_{m,n}$:

$$w_{m,n} = (\text{число в предыдущем окне}) - (\text{число ушедших}) + (\text{число присоединенных}).$$

Практическое применение. Метод свертки с вычислением разностей успешно применен в двух приложениях — анализ документа, отсканированного на высоком разрешении, и распознавание биологических клеток по изображению, поступающему с микроскопа. С помощью настоящего метода получены быстрые алгоритмы для следующих операций: фильтрация изображения, требующая вычисления средних (например, фильтр Винера) в области, задаваемой одновременно бинарным ядром и бинарной маской; поиск и связывание соседних объектов; вычисление признаков объектов; ранговая фильтрация[1] (в частности, медианная).

[1]Она не является интегральным преобразованием, но подход остается тот же. Сортировка вышедших/вошедших элементов и последующее сливание с уже отсортированной последовательностью большого окна оказывается более эффективной, чем сортировка всех элементов большого окна.

В табл. 1 сравниваются алгоритмы свертки в условиях, приведенных в табл. 2. Разностный алгоритм в 5,6 раз быстрее непосредственной свертки, но в 4,7 раза уступает быстрой свертке на базе БПФ. С другой стороны, преимуществами разностного подхода является точность, локальность и отсутствие необходимости в вычислениях с плавающей точкой и в дополнительной памяти (для примера в табл. 2 — 64 Мб). Это сравнительно увеличивает реальное быстродействие разного алгоритма.

Таблица 1

Сравнительная характеристика алгоритмов

Алгоритм	Число операций		Дополнительная память
	абсолютное	относительное	
Непосредственная свертка (квадрат)	$MN \cdot (2R)^2 = 6{,}712 \cdot 10^9$	1,319	0
Непосредственная свертка (кольцо)	$MN\pi \left(R^2 = R^2_{\text{внеш}}\right) = 5{,}086 \cdot 10^9$	1,000	0
Быстрая свертка (БПФ)	$2MN \log_2 (MN) + MN = 1{,}887 \cdot 10^8$	0,037	$8{\cdot}2{\cdot}MN =$ 64 Мб
Разностная свертка	$MN{\cdot}2\pi \left(R + R_{\text{внеш}}\right) = 8{,}960 \cdot 10^8$	0,176	0

Таблица 2

Параметры свертки

Размер исходного изображения	$M = N = 2048$
Вид ядра	бинарное кольцо с $R_{\text{внутр}} = 14$ и $R_{\text{внеш}} = R = 20$
Размер окна	$2R \times 2R = 20^2$
Число байт на пиксел	1
Шаг при движении окна	1

Выводы. Рассмотрен метод интегрального преобразования, основанный на вычислении разности между соседними суммами. Хотя разностный подход достаточно очевиден, его приложение к обработке изображений ранее не исследовалось. Предложенный метод преобразования изображения эффективен, когда необходимо выполнить точное суммирование внутри ограниченной области, заданной маской или другим способом. То есть, приложение этого метода оправдано для бинарных ядер простой формы (круг, кольцо, лепестки и т.п.) и не целесообразно для неоднородных ядер ($\sin c$, вейвлеты, гауссовская функция).

СПИСОК ЛИТЕРАТУРЫ

1. Гольденберг Л. М. Цифровая обработка сигналов / Л.М. Гольденберг, Б.Д. Матюшкин, М.Н. Поляк. – М.: Радио и связь, 1985.
2. The Scientist and Engineer's Guide to Digital Signal Processing. Second Edition / Ed. by S. W. Smith. – California Technical Publishing, 1999.. http://www.dspguide.com/
3. Blahut R. E. Fast algorithms for digital signal processing / R.E. Blahut. – Addison-Wesley, 1985.
4. Numerical Recipes in C. The Art of Scientific Computing. Second Edition / W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. – Cambridge University. Press, 1988. http://www.library.cornell.edu/nr/
5. Arndt, J. Algorithms for programmers: ideas and source code. – 2002. http://www.jjj.de/

Статья поступила в редакцию 5.07.2005

УДК 681.3:519.6

Г. С. Иванова

ФОРМАЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ СТРУКТУРИЗАЦИИ АЛГОРИТМОВ

Проанализированы характеристические особенности структурных алгоритмических конструкций, предложены математические модели как самих конструкций, так и структурных алгоритмов в целом. Определено понятие факторизации фрагмента алгоритма и аксиоматика этой операции. Выявленные свойства структурного алгоритма и формальная постановка задачи структуризации неструктурного являются основой для автоматизированного решения задачи структуризации алгоритмов.

Одним из принципов построения технологичных программ считается структурное программирование [1], которое предполагает при разработке программ на универсальных процедурных языках высокого уровня использование только так называемых "структурных" конструкций. К структурным относят три основные конструкции: следование, ветвление и цикл с условием на входе ("цикл-пока"), и три дополнительные: выбор, цикл с заданным количеством повторений ("счетный") и цикл с условием на выходе ("цикл-до"). Из накопленного к настоящему времени опыта структурного программирования следует, что технологичность программ, составленных с использованием только этих структур, гораздо выше, чем в том случае, когда разработчик использует неструктурное программирование. Это связано с тем, что структурные программы легче анализировать, они содержат меньше ошибок, проще тестируются и отлаживаются. Однако