

О быстром поиске сумм по сегментам за счет предподсчета Хаф-пирамид

Константин Сошин^{1,2}, Егор Ершов¹ и Сергей Гладилин¹

¹ Институт проблем передачи информации им. А.А. Харкевича, Москва, Россия,

² Московский физико-технический институт (НИУ), Долгопрудный, Россия
soshin@visillect.com

Аннотация В работе предложен алгоритм быстрого вычисления суммы по произвольному отрезку изображения, заданному парой концов, трудоёмкость которого составляет $\Theta(\log n)$ операций, где n – линейный размер изображения. Идея алгоритма состоит в использовании, так называемой БПХ-пирамиды, полученной в результате сохранения промежуточных результатов вычисления быстрого преобразования Хафа, предложенного П. Бреди в 1992 году. Предподсчет такой пирамиды требует $\Theta(n^2 \log n)$ операций.

Keywords: Диадический паттерн, быстрое преобразование Хафа, трехмерное быстрое преобразование Хафа, пирамида отрезков.

1 Введение

В работах Донохо и соавторов [1] предложен подход к многомасштабному анализу изображений, названный *бимлет*-анализом. Подход предполагает построение *словаря бимлетов* как диадически организованного набора отрезков на изображении всех диадических положений и длин, имеющих всевозможные ориентации. *Бимлет-преобразование* – преобразование суммирования изображения по всем бимлетам в словаре – позволяет построить по изображению $n \times n$ пикселей «бимлет-пирамиду», содержащую $\Theta(n^2 \log n)$ значений. Предложенные Донохо и соавторами алгоритмы анализа этой пирамиды позволяют строить детекторы объектов (отрезков неизвестного положения, длины и ориентации, кругов неизвестного диаметра и т.д.) на сильно зашумленном изображении [2].

Однако авторы опускают вопрос того как именно быстро рассчитать суммы по всем бимлетам пирамиды, предполагая, что они уже посчитаны. В то же время Донохо и соавторы перечисляют несколько исследований, направленных на быстрое вычисление дискретного преобразования Радона (также известного как преобразование Хафа). Один из этих подходов – предложенный Бреди [3] – сейчас известен как быстрое преобразование Хафа (БПХ) [4]. Он позволяет за $\Theta(n^2 \log n)$ операций точно вычислить полный Хаф-образ изображения по диадическим паттернам, хорошо аппроксимирующим прямые на изображении. Ортогональная ошибка аппроксимации прямой диадическим паттерном составляет $\frac{\log_2 n}{6}$ [5].

В настоящей работе на основе предподсчета пирамиды диадических подпаттернов БПХ (БПХ-пирамиды), требующего $\Theta(n^2 \log n)$ операций, предлагается алгоритм быстрого (за $\Theta(\log n)$) поиска суммы вдоль произвольного отрезка диадического паттерна на изображении, заданного парой точек. Данный алгоритм, с одной стороны, является способом быстрого построения приближенной пирамиды бимлетов (которое можно осуществить за $\Theta(n^2 \log^2 n)$ операций), а с другой – имеет самостоятельную ценность в задачах, требующих вычислять многочисленные суммы по отрезкам, но не требующих столь высокоуровневой конструкции, как бимлет-анализ.

Также в работе рассмотрена проблема быстрого подсчета сумм для сегментов прямых и плоскостей в трехмерном изображении.

2 Быстрая суммация по сегментам прямых

Рассмотрим способ быстрого вычисления суммы по любому отрезку на изображении с использованием предподсчитанной БПХ-пирамиды. Предподсчет выполняется в процессе вычисления быстрого преобразования Хафа от изображения. В процессе его вычисления происходит сначала суммация по диадическим паттернам длиной 2 пикселя, затем из пар таких сумм собираются суммы по диадическим паттернам длиной 4 пикселя, из них суммы по паттернам длиной 8 пикселей и т.д. Если сохранять все промежуточные суммы, то формируется БПХ-пирамида, которая содержит суммы по всем диадическим паттернам длины 2^k , где $k \in \overline{1, \log_2 n}$, а n – линейный размер изображения³, выровненным по линейному диадическому разбиению изображения [7]. Размер хранимых паттернов 2^k соответствует номеру уровня пирамиды k .

Трудоёмкость вычисления БПХ и, соответственно, размер БПХ-пирамиды составляют $\Theta(n^2 \log n)$, т.е. каждый элемент пирамиды вычисляется за константное время. Интересно заметить, что вычислить сумму по всем диадическим прямым на изображении быстрее чем за $\Theta(n^2 \log n)$ невозможно [6].

Алгоритм поиска суммы по отрезку, заданному концами, разделим на два этапа: 1. поиск диадической прямой, содержащей этот отрезок; 2. поиск суммы по отрезку на прямой. В данной работе рассматривается задача 2 и опускается задача 1.

Оптимальный алгоритм нахождения суммы по некоторому отрезку AB на диадической прямой CD ($AB \subset CD$, точки C и D – на границе изображения) заключается в разбиении AB на минимальное количество отрезков, входящих в БПХ-пирамиду. Суммы по каждому из них уже известны (в результате предподсчета пирамиды), поэтому требуется только сложить их. Однако авторам не известны реализации данного алгоритма. В настоящей работе предлагается такой же по асимптотической трудоёмкости, но несколько более простой в реализации алгоритм, в котором искомая сумма

³ В нашей работе мы будем считать, что сторона изображения n равна степени двойки.

представляется как разность сумм по отрезкам CB и CA , сумма по каждому из которых в свою очередь вычисляется путем разбиения на отрезки, входящие в БПХ-пирамиду (рис. 1).

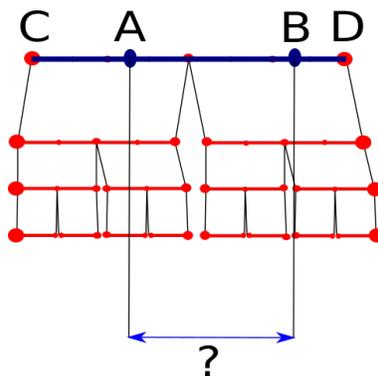


Рис. 1. Иллюстрация задачи поиска произвольного отрезка AB на прямой CD . Красным показаны отрезки, сумма по которым известна (содержится в Хаф-пирамиде).

Опишем способ нахождения суммы вдоль отрезка между точкой A диадического паттерна и левой граничной точкой C . Алгоритм, по-существу, представляет из себя бинарный поиск разложения отрезка AC на множество предподсчитанных отрезков. Инициализируем аккумулятор нулем. Разделим отрезок CD вертикальной прямой пополам. Если точка A лежит правее проведенной прямой, то увеличим аккумулятор на значение суммы на подпаттерне, находящемся в левой половине (она нам известна из БПХ-пирамиды). Далее повторяем эту процедуру для той половины, где лежит точка A . На каждой итерации мы увеличиваем аккумулятор на сумму по левому подпаттерну, если точка A находится правее разделяющей прямой и не изменяем его, если точка лежит левее. Таким образом мы находим сумму от граничной точки C изображения до данной A за $\Theta(\log n)$. Аналогично находим сумму вдоль CB , а, вычтя из одной суммы другую, мы получим искомую сумму по отрезку AB .

Заметим, что если к бинарному разложению координаты x_a точки A (где C имеет нулевую координату, а D – координату $n - 1$) прибавить единицу, то оно непосредственно кодирует набор предподсчитанных отрезков из которых складывается искомый: k -й бит определяет наличие следующего подотрезка длинны 2^k . Соответственно, алгоритм заключается в побитовом проходе этого разложения от старшего бита к младшему, что, очевидно, выполняется за $\Theta(\log n)$.

Обобщение предлагаемого алгоритма на трехмерный случай возможно двумя способами: можно ставить задачу быстрого вычисления суммы по

прямой или по плоскости на трехмерном изображении. Первый случай близок к двумерному: в процессе вычисления трехмерного БПХ для прямых [7] необходимо запоминать промежуточные результаты суммирования и формировать БПХ-пирамиду. Схема нахождения суммы по любому отрезку паттерна при этом полностью аналогична двумерному случаю. Второй же случай требует отдельного рассмотрения.

3 Быстрая суммация по сегментам плоскостей

Как и в двумерном случае, будем считать, что задача поиска конкретной диадической плоскости $EFGH$ решена, и требуется найти сумму по некоторому параллелограмму $ABCD$ этой плоскости (смотри рисунок 2). Будем рассматривать только такие параллелограммы, стороны которых параллельны границам содержащих их плоскостей. Здесь и далее будем смотреть на $EFGH$ как на двумерное изображение, а на $ABCD$ – как на прямоугольник, содержащийся в ней.

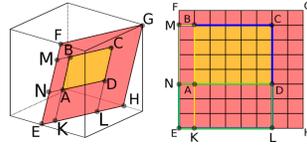


Рис. 2. Плоскость $EFGH$, содержащая сегмент $ABCD$, и их двумерное изображение.

Решение задачи сведем к задаче нахождения сумм по прямоугольникам, левый нижний угол которых лежит в точке E самой дискретной плоскости. Откуда по формуле включения-исключения найдем сумму по $ABCD$:

$$\Sigma_{ABCD} = \Sigma_{EMCL} - \Sigma_{EMBK} - \Sigma_{ENDL} + \Sigma_{ENAK}$$

Приведем алгоритм нахождения суммы по прямоугольнику $EMCL$. Аналогично двумерному случаю будем считать, что изображение $ABCD$ имеет $\log_2 n$ разбиений на квадраты со стороной $\frac{n}{2^k}$, где $k \in \overline{1, \log_2 n}$ (смотри рисунок 3). Для каждого элемента каждого разбиения значение суммы по нему считаем известным. Алгоритм быстрого подсчета этих сумм будет изложен в приложении.

Нам требуется покрыть наименьшим числом предподсчитанных квадратов искомый прямоугольник $EMCL$. Поскольку он лежит своей левой нижней вершиной в точке E дискретной плоскости, мы начнем его заполнение с наибольшего квадрата, содержащегося в нем, левая нижняя вершина которого тоже лежит в точке E .

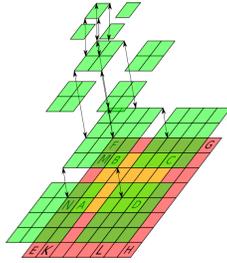


Рис. 3. Иллюстрация задачи поиска суммы по параллелограмму при условии наличия предподсчета. Зеленым показаны квадраты, суммы по которым мы знаем.

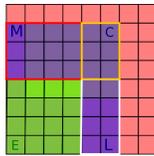


Рис. 4. Иллюстрация работы алгоритма суммации по *EMCL*. Зеленым показано покрытие наибольшим квадратом. Фиолетовым показана рамка. Красным обведена верхняя часть рамки, белым – правая, а желтым – угловая.

После покрытия углового квадрата, останется часть, которую назовем рамкой. Разделим ее на верхнюю, правую и угловую части (смотри рис. 4). Заполнение правой части происходит слева направо столбцами размера $l \times 2^k$, где k последовательно принимает значения $\{\log_2 l, \dots, 1, 0\}$, а l – это сторона большого углового квадрата. Каждый из столбцов размеров $l \times 2^k$ состоит из $\frac{l}{2^k}$ предподсчитанных квадратов размером $2^k \times 2^k$ (см рисунок 5).

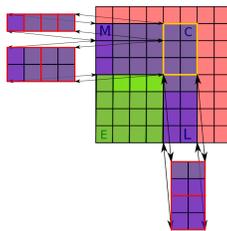


Рис. 5. Иллюстрация покрытия столбцами из предподсчитанных квадратов правой и верхней частей рамки. Зеленым показано покрытие наибольшим квадратом. Желтым обведена угловая часть рамки, которая вычисляется на следующей итерации.

Аналогичным образом заполняется верхняя часть. К оставшейся неподсчитанной угловой части рекурсивно применим описанный выше алгоритм. А именно, точно так же покроем наибольший левый нижний квадрат и посчитаем верхнюю и правую части уже новой рамки. Будем продолжать алгоритм, пока на очередной итерации угловая часть не будет состоять из единственного пикселя.

Оценим вычислительную сложность приведенного способа. На очередной итерации вычисление правой или верхней части требует не более чем $\sum_{k=0}^{\log_2 n - 1} 2^k = n - 1$ операций. Тогда всего на вычисление рамки без угловой части требуется не более $2n - 3$ операций. Итераций алгоритма не более, чем $\log_2 n$. Следовательно, трудоемкость приведенного алгоритма составляет $O(n \log n)$.

4 Приложение

Приведем алгоритм вычисления трехмерного быстрого преобразования Хафа для плоскостей, используемого в предподсчете. Введем типизацию плоскостей в трехмерном изображении как это сделано в работе [7]. Будем называть плоскость преимущественно-перпендикулярной к оси, если нормаль к ней образует с данной осью меньший угол, чем с двумя другими. Будем рассматривать только плоскости, преимущественно-перпендикулярные к оси x . Преимущественно-перпендикулярные разделяются на 4 типа наклонов, как это показано на рисунке 6.

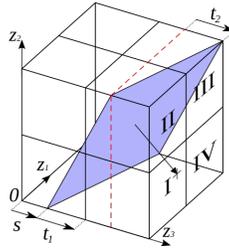


Рис. 6. Иллюстрация типизации плоскостей в пространстве. Картинка взята из работы [7]

Без ограничения общности считаем, что рассматриваемые плоскости имеют тип I.

Разделим изображение на массивы пикселей размером $n \times 2 \times 2$, параллельные оси x , как показано на рисунке 7. Обозначим четыре пиксельные строки в каждом массиве следующим образом: нижняя строка, ближайшая к x обозначим как a ; нижняя, дальняя от x — как b ; верхняя, ближайшая к x

– как c ; верхняя, дальняя от x – как d . Просуммируем поэлементно строки a, b, c, d в каждом массиве. Это даст нам суммы по квадратным подпаттернам из 4 пикселей с нулевыми наклонами (1-ый тип на рисунке 7). Сделаем циклический сдвиг строк c и d вдоль положительного направления оси x в каждом массиве на исходном изображении и опять выполним поэлементную суммирование. Таким образом получим суммы по всем подпаттернам с наклоном 2-го типа. Циклический сдвиг b и d даст наклон 3-го типа, а сдвиг c и b на 1 и сдвиг d на 2 даст подпаттерны с наклоном 4-го типа (рисунок 7) соответственно.

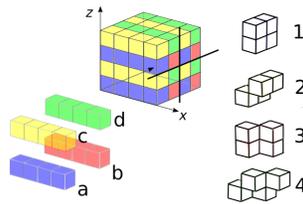


Рис. 7. Первая итерация ТБПХ для плоскостей на изображении $4 \times 4 \times 4$. Показано обозначение строк в каждом массиве $n \times 2 \times 2$. Показаны подпаттерны, суммы по которым вычисляются.

Полученное преобразованное изображение используется на следующем шаге рекурсии вычисления суммы по подпаттернам размером в 16 пикселей в каждом массиве $n \times 4 \times 4$, состоящего из четырех предподсчитанных подмассивов $n \times 2 \times 2$. Так, чтобы вычислить суммы по паттернам из 16 пикселей с нулевым наклоном как у синего паттерна на рисунке 8, надо поэлементно сложить суммы по подпаттернам типа 1 из каждого подмассива $n \times 2 \times 2$. Процедура продолжается, пока размер результирующего изображения не станет равным $n \times n \times n$.

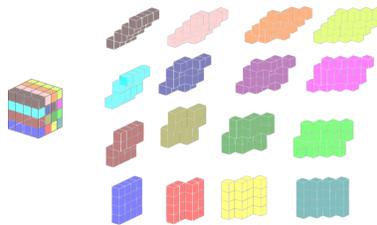


Рис. 8. Вторая итерация трехмерного БПХ для плоскостей на изображении $4 \times 4 \times 4$. Суммирование происходит по всем паттернам из 16 пикселей.

5 Заключение

В работе предложен метод суммации по произвольным отрезкам известного паттерна на двумерных и трехмерных изображениях с использованием предподсчета. Показано, что трудоемкость метода составляет $\Theta(\log n)$. Так же приведен алгоритм нахождения суммы по произвольным параллелограммам со сторонами, параллельными границам изображения, лежащим в заданной дискретной плоскости на трехмерном изображении, имеющий вычислительную сложность $O(n \log n)$. Для предподсчета предложена реализация трехмерного быстрого преобразования Хафа для плоскостей.

Благодарности

Коллектив авторов отдельно благодарит Дмитрия Петровича за постановку задачи, а также плодотворные беседы по теме данной работы. Исследование выполнено при частичной финансовой поддержке РФФИ в рамках научного проекта №17-29-03514.

Список литературы

1. D. L. Donoho, X. Huo: Beamlets and Multiscale Image Analysis. Multiscale and multiresolution methods. Springer. 149-196 (2002).
2. Arias-Castro, E., Donoho, D. L., Huo, X: Near-optimal detection of geometric objects by fast multiscale methods IEEE Trans. on Inf. Theory, 51(7), 2402–2425 (2005)
3. Brady M. L., Yong W.: Fast parallel discrete approximation algorithms for the Radon transform. Proc. of the 4th ann. ACM symp. on Par. alg. and arch. ACM. 91-99. (1992)
4. Nikolaev D. P. et al. Hough transform: underestimated tool in the computer vision field //Proceedings of the 22th European Conference on Modelling and Simulation. – 2008. – Т. 238. – С. 246. (2008)
5. E. Ershov, S. Karpenko: Fast Hough Transform and approximation properties of dyadic patterns. arXiv preprint arXiv:1712.05615 (2017)
6. Timur M. Khanipov Computational complexity lower bounds of certain discrete Radon transform approximations <https://arxiv.org/abs/1801.01054>
7. Ершов Е. И., Терехин А. П., Николаев Д. П. Обобщение быстрого преобразования Хафа для трехмерных изображений //Информационные процессы. – 2017. – Т. 17. – №. 4. – С. 294-308.
8. Ershov E., Terekhin A., Karpenko S., Nikolaev D., Postnikov V. Fast 3D Hough Transform computation. Proceedings 30th European Conference on Modelling and Simulation, 2016, pp. 227-230.