Информационные процессы, Том 17, № 4, 2017, стр. 294–308

МАТЕМАТИЧЕСКИЕ МОДЕЛИ, ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

Обобщение быстрого преобразования Хафа для трехмерных изображений [1]

Е.И. Ершов, А.П. Терехин и Д.П. Николаев

Институт проблем передачи информации, Российская академия наук, Москва, Россия

Поступила в редколлегию 31.08.2017

Аннотация—Работа посвящена исследованию алгоритмов вычисления быстрого преобразования Хафа для двумерных и трехмерных изображений. Предложен метод вычисления быстрого преобразования Хафа (БПХ) для прямых в трехмерном изображении, асимптотическая сложность и объем требуемой памяти которого составляют $\Theta(n^4)$, где n – характерный линейный размер исходного изображения. Рассматриваются алгоритмы БПХ для аппроксимации в двумерном и трехмерном пространствах, исследуются свойства точности и полноты соответствующих множеств диадических паттернов.

КЛЮЧЕВЫЕ СЛОВА: дискретное преобразование Радона, дискретное преобразование Йона, трехмерное преобразование Хафа, быстрое преобразование Хафа, точность преобразования Хафа, трудоемкость преобразования Хафа

Преобразование Хафа (ПХ) было предложено в 1959 году Полом Хафом в качестве инструмента анализа фотографий следов частиц в пузырьковой камере [1]. ПХ ставит в соответствие каждому паттерну из определенного семейства (прямые, окружности, эллипсы) значение, равное сумме (или результату любой другой операции, заданной на множестве) значений пикселей изображения, принадлежащих этому паттерну. Например, в случае двумерного изображения, преобразование Хафа для прямых (а точнее – дискретных паттернов, аппроксимирующих соответствующие геометрические прямые) вычисляет суммы по всем дискретным прямым заданной параметризации и структуры. Термин “дискретный паттерн” здесь обозначает определённое множество пикселей. В данной работе речь пойдет именно о таком понимании ПХ и двух его обобщениях для трехмерных изображений. В целях лаконичности будем употреблять термин “преобразование Хафа”, подразумевая преобразование Хафа для дискретных прямых на плоском изображении. Стоит оговориться, что во многих источниках такое преобразование также носит название дискретного преобразования Радона [2, 3].

Существенным недостатком ПХ является высокая асимптотическая сложность: $\Theta(n^3)$ операций, где n – линейный размер изображения. Данное обстоятельство на практике приводит к необходимости создания схем ускоренного подсчета ПХ. Среди работ по созданию вычислительно эффективных схем его подсчета можно выделить два различных подхода: первый – это поиск и оценивание значений локальных максимумов в пространстве Хафа, а второй – это создание эффективного алгоритма вычисления всего Хаф-образа (или его аппроксимации).

Классическими представителями первого подхода являются различные рандомизированные техники выполнения процедур голосования [4–6], где прямая (или плоскость) оценивается по количеству лежащих близко к ней точек. Основным достоинством таких алгоритмов является малое время работы, однако, очевидным недостатком – нестабильность работы при одних и тех же входных данных и параметрах.

В работе [7] предложен алгоритм, решающий проблему недетерменированного результата. По существу, предложенная схема является комбинацией алгоритмов кластеризации точек

[1] Работа выполнена при финансовой поддержке грантов РФФИ №15-29-06079, №17-29-03297

исходного пространства и процедур голосования для наиболее выделяющихся кластеров. Важной особенностью алгоритма является зависимость вычислительной сложности от числа точек k, а не от размера изображения, и составляет $O(k \log k)$.

Правда, в задачах, где входом является не облако точек, а массив значений, выполнение этого алгоритма в режиме реального времени невозможно уже для трехмерного изображения с линейным размером 128 пикселей [7]. Иными словами, алгоритм удобен для использования только в случае разреженных входных данных. Стоит также отметить, что предложенная процедура фильтрации данных приводит к неочевидным искажениям соответствующего им распределения в пространстве Хафа, в котором и производится поиск плоскости с наибольшей суммой. Более того, формально алгоритмы, отнесенные к первому подходу, не являются реализациями преобразования Хафа, так как они не вычисляют Хаф-образ, хотя и позволяют оценить параметры прямых с наибольшей суммой на изображении.

Одним из представителей второго подхода является эффективный алгоритм вычисления аппроксимации преобразования Хафа – быстрое преобразование Хафа (БПХ), в некоторых работах он носит название быстрого дискретного преобразования Радона. Судя по всему, данный алгоритм был независимо открыт не менее пяти раз [8–12]. Вычислительная сложность данного алгоритма существенно снижена по сравнению с ПХ до $O(n^2 \log n)$ операций. Особенностью алгоритма является использование диадических дискретных паттернов, структура которых позволяет оптимизировать вычисления, но характеризуется неточностью аппроксимации соответствующей геометрической прямой. Иными словами, БПХ вычисляет обобщенное преобразование Хафа для диадических паттернов, которое для классического ПХ является аппроксимацией. Неудивительно, что не прекращаются попытки по созданию генераторов точных быстрых вычислительных схем ПХ [20], но достаточно быстрого алгоритма в общем виде не построено до сих пор. Другой особенностью БПХ является то, что он позволяет заменять сложение любой другой ассоциативной операцией, например, вычислением максимума.

В настоящее время алгоритм вычисления БПХ востребован для решения множества практических задач как на ПК, так и на маломощных встраиваемых системах. Существует множество применений БПХ, таких как детектирование прямолинейных краев [11], в частности, определение границ документа [13] и детекция дорожной разметки [14], определение точек схода [15], в качестве составной части алгоритма для томографического восстановления [16], для реализации алгоритмов слепой калибровки радиальной дисторсии [17].

В последние же годы фокус внимания в области анализа изображений сместился в сторону трехмерных изображений, что связано с ростом популярности их источников: в медицине это аппараты томографии и трехмерного УЗИ, в робототехнике – это различные сенсоры глубины (стереопары, лазерные дальномеры, камеры со структурированным светом и прочее). В связи с этим возникает потребность в эффективных схемах вычисления ПХ для трехмерных изображений.

Так, в работе [18] был предложен алгоритм трехмерного быстрого преобразования Хафа (ТБПХ) для плоскостей. Показано, что асимптотическая вычислительная сложность данного алгоритма составляет $\Theta(n^3 \log n)$ операций (n – линейный размер равностороннего трехмерного изображения), независимо от заполненности пространства. Для сравнения, асимптотическая сложность наивного алгоритма вычисления преобразования Хафа в этом случае составляет $\Theta(n^5)$ операций.

В настоящей работе предлагается алгоритм ТБПХ для прямых, приведены асимптотические оценки его вычислительной сложности, на основе которых показана асимптотическая оптимальность данного алгоритма для последовательных архитектур. Помимо этого в работе исследуются полнота и точность аппроксимации систем диадических паттернов для плоского и трехмерного изображений.

1. БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ХАФА ДЛЯ ДВУМЕРНОГО СЛУЧАЯ

В разделе предлагается описание БПХ с особым вниманием к конкретным свойствам алгоритмической схемы, которые существенны при создании алгоритмов ТБПХ для прямых и для плоскостей. Помимо этого, в данном разделе доказано утверждение, что для любой пары пикселей на двумерном изображении найдется хотя бы один диадический паттерн через них проходящий. Определим изображение как многомерный (в данной работе двух-, трехмерный) массив данных. Для простоты изложения далее будем рассматривать только равносторонние изображения с линейным размером $n = 2^m$, где m – натуральное число.

1.1. Параметризация и паттерны БПХ

Введем (s, t)-параметризацию прямой на изображении, по существу она аналогична классической параметризации угловым коэффициентом ($y = kx + b$). Пусть начало системы координат и координатные оси располагаются так, как показано на рисунке 1. Введем два типа геометрических прямых: преимущественно-горизонтальные прямые (ПГП) и преимущественно-вертикальные прямые (ПВП). Первые в параметризации угловым коэффициентом лежат в диапазоне $|k| \leq 1$, а вторые – $|k| \geq 1$.

ПГП будем задавать параметрами s и t координат двух принадлежащих ей точек на вертикальных границах изображения: $(0, s)$ и $(n-1, s+t)$, а ПВП – на горизонтальных: $(s, 0)$ и $(s+t, n-1)$, где $t \in [-n+1; n-1]$ и $s \in [-n+1; n-1]$. Знак t определяет направлением относительно координатной оси. Если $t \leq 0$, то такие прямые называются ПГП с наклоном вниз, если $t \geq 0$ – с наклоном вверх. Краткости ради дальнейший материал будет изложен для ПГП с наклоном вверх. Одним из преимуществ (s, t)-параметризации по сравнению с параметризацией угловым коэффициентом является то, что преобразование Радона изображения в такой параметризации не равно нулю на ограниченной области, равно как и ПХ – его дискретизация. Именно это и позволяет задавать границы Хаф-образа в пространстве Хафа.

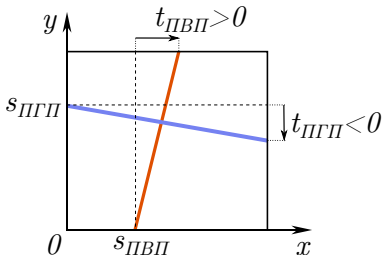

Рис. 1. Иллюстрация (s, t)-параметризации. Синяя прямая принадлежит классу преимущественно-горизонтальных с наклоном вниз, красная – преимущественно-вертикальным с наклоном вправо.

В данной работе мы рассмотрим два разных дискретных паттерна, аппроксимирующих прямую: диадический и ближайший. Ближайший паттерн известен благодаря работе Брезенхема [19], посвященной его построению без умножений и делений. В каждом столбце изображения ему принадлежит один пиксель, центр которого ближе всего располагается к аппроксимации геометрической прямой. В аналитическом виде формула вычисления координат пикселей, принадлежащих ближайшему паттерну, может быть записана следующим образом:

$$y = \left[\frac{tx}{n-1} + s\right]. \tag{1}$$

Далее в работе без ограничения общности при описании свойств паттернов будем полагать, что $s = 0$. Из определения ближайшего паттерна ясно, что ошибка его аппроксимации орди-

наты в каждом столбце решетки не превышает половины размера пикселя (равна 1/2). Будем называть такую ошибку *ортотропной* и вычислять по формуле:

$$E^b_{t,n}(x) = \left| \frac{tx}{n-1} - \left[\frac{tx}{n-1} \right] \right|, \tag{2}$$

где символ b обозначает, что ошибка рассчитывается именно для ближайшего паттерна. Рассмотрим теперь свойства *ортогональной* $N_{t,n}(x)$ ошибки аппроксимации (расстояния от центра пикселя до геометрической прямой). Заметим, что для любого центра пикселя его ортогональная и ортотропная $E_{t,n}(x)$ ошибки относительно прямой с наклоном t связаны выражением:

$$N_{t,n}(x) = \frac{E_{t,n}(x)}{\sqrt{1 + (\frac{t}{n-1})^2}}. \tag{3}$$

Следовательно, максимальная ортогональная ошибка аппроксимации ограничена сверху максимальной ортотропной. Покажем, что для ближайших паттернов 1/2 – точная верхняя грань ортогональной ошибки. Рассмотрим прямую с наклоном $t = 1$ и заметим, что для пикселя с координатами $(n/2, 1)$, при стремлении n к бесконечности, ортотропная ошибка будет стремиться к 1/2, а угол между направлениями ошибок будет уменьшаться, в результате чего ортогональная ошибка будет стремиться к ортотропной снизу.

Однако для ближайшего паттерна не известно алгоритма вычисления быстрого преобразования Хафа (БПХ). Существуют, впрочем, работы по созданию компилятора обобщенных преобразований Хафа, параметризуемого типом паттерна, который для ближайших паттернов на изображениях малого размера позволяет получить вычислительную схему, сопоставимую по сложности с БПХ [20,23]. Но в них не рассматривается возможность компиляции алгоритма быстрого вычисления для изображений с линейным размером более 128 пикселей.

При создании алгоритма БПХ был предложен так называемый диадический паттерн (ДП). Существует два способа задавать ДП – рекурсивный и аналитический, и, поскольку в данной работе они нам потребуются, изложим оба.

Рекурсивное определение записывается следующим образом:

$$P^n_t = P^{n/2}_{\lfloor t/2 \rfloor} \cup \left(P^{n/2}_{\lfloor t/2 \rfloor} \nearrow (n/2, \lfloor t/2 \rfloor + t \bmod 2) \right), \tag{4}$$

где P^j_i – множество пикселей, принадлежащих паттерну с наклоном i и длиной j, а стрелка обозначает приращение координат всех элементов множества на указанный вектор, причем $P^1_0 = \{(0,0)\}$. Опишем подробнее, что именно за правило описывает эта формула. На каждом шаге рекурсии изображение делится вертикально на две равные части, и для каждой из этих частей определяется наклон диадического подпаттерна и сдвиг (0 или 1) между ними. Эти значения задают координаты пикселей на границе разделения: $(n/2 - 1, \lfloor t/2 \rfloor)$ и $(n/2, \lfloor t/2 \rfloor + t \bmod 2)$. Затем процедура проводится для каждого подпаттерна до тех пор, пока длина изображения не будет равна единице. Из определения видно, что диадический паттерн обладает самоподобной структурой. Используя данное определение, можно показать связь между битовым представлением наклона t и структурой паттерна. Если в битовом представлении в k-м разряде стоит единица, значит на k-м шаге рекурсии каждая пара подпаттернов будет сдвинута на единицу относительно друг друга.

Аналитический способ задания диадического паттерна выглядит следующим образом [24]:

$$D^n_t(x) = \sum_{k=0}^{p-1} t_k \left[\frac{2^k x}{n-1} \right], \tag{5}$$

где t_k – значения k-го разряда в битовом представлении наклона t, $D_t^n(x)$ – значение ординаты пикселя диадического паттерна. Таким образом, выражение $D_t^n(x) = y$ эквивалентно выражению $(x, y) \in P_t^n(x)$.

Показано [24], что ортотропная ошибка аппроксимации ДП, вычисляемая по формуле

$$E_{t,n}^d(x) = \left| \frac{tx}{n-1} - D_t^n(x) \right|, \tag{6}$$

ограничена сверху значением $\frac{m}{6}$, достигающимся при четных m при $t_1 = (n-1)/3$, $t_2 = 2(n-1)/3$.

Для диадического паттерна, аналогично ближайшему, верхняя оценка для максимальной ортогональной ошибки аппроксимации мажорируется ортотропной и составляет $\frac{\log_2 n}{6}$ пикселей. Оценим максимальную ортогональную ошибку снизу, взяв наибольшую известную нам достижимую ошибку. Для этого рассмотрим наклон t_1, в этом случае угол между ортогональным и ортотропным направлениями равен $\alpha = arctg(\frac{t_1}{n-1})$. Ортогональная ошибка $N_{t,n}^d = \frac{\log_2 n}{6} \cdot \cos\alpha$ при четном m равна $\frac{\log_2 n}{2\sqrt{10}}$, что и составляет оценку максимальной ортогональной ошибки снизу.

1.2. Алгоритм быстрого преобразования Хафа для двумерного случая

Перейдем непосредственно к описанию алгоритма БПХ для преимущественно-горизонтальных паттернов с наклоном вверх ($t \in [0, n-1]$). Входом алгоритма является двумерное изображение, а выходом – двумерный Хаф-образ, в каждом пикселе которого содержится значение суммы по соответствующему координатам этого пикселя диадическому паттерну. Код функции $fht2$ для вычисления БПХ представлен ниже на языке MATLAB.

```
function h = fht2(I)
  n = size(I, 2);
  if n < 2
    h = m;
    return
  end
  h = mergeHT(fht2(I(:, 1 : n / 2, :)), fht2(I(:, (n / 2 + 1) : n, :)));
end

function h = mergeHT(h0, h1)
  [m, n0, o] = size(h0);
  n = 2 * n0;
  h = zeros(m, n, o);
  r = (n0 - 1) / (n - 1);
  for i = 1 : n
    t = i - 1;
    t0 = round(t * r);
    s = t - t1;
    h(:, i, :) = h0(:, t0 + 1, :) + [h1(s + 1 : m, t0 + 1, :); h1(1 : s, t0 + 1, :)];
  end
end
```

Листинг 1. Алгоритм быстрого преобразования Хафа для преимущественно-горизонтальных прямых с наклоном вверх, записанный на языке MATLAB.

Входное изображение I рекурсивно разбивается на два равных изображения вертикальной прямой, пока ширина его не будет равна 2. Затем каждый столбец суммируется с парным поэлементно, после чего правый столбец циклически сдвигается вниз на один пиксель и выполняется второе поэлементное суммирование. В результате вычисляются суммы по паттернам

длины 2 с наклонами: $t = 0$ и $t = 1$. Полученные суммы используются на следующем шаге рекурсии для вычисления сумм по паттернам длины 4 в каждой четверке столбцов. Так, чтобы вычислить все суммы по паттернам длины 4 и наклоном $t = 0$, достаточно сложить суммы по паттернам длины 2 у соседних пар столбцов. Процедура продолжается, пока ширина результирующего изображения не будет составлять n.

Оценим вычислительную сложность БПХ. Сумма по всем паттернам длины 2 для пары столбцов требует $2n$ операций, и таких пар – $n/2$, для следующего уровня рекурсии требуется $4n$ операций, но четверок на изображении в два раза меньше, чем пар – $n/4$, продолжая, получим следующую сумму $2n \cdot \frac{n}{2} + 4n \cdot \frac{n}{4} + ... + n^2$, а в силу того, что таких слагаемых $\log_2 n$ (по числу уровней рекурсии), окончательно получим оценку сложности $n^2 \log_2 n$. Соответственно, у БПХ для всех типов прямых суммарная сложность будет составлять $4n^2 \log_2 n$.

Ясно, что вычисление БПХ может быть ускорено за счет параллелизации вычислений. Поскольку на каждом шаге алгоритма выполняется в точности n^2 независимых сложений, то с использованием n^2 процессоров данные вычисления можно провести за константное время. А так как глубина рекурсии равна $\log_2 n$, то время вычисления может быть в этом случае оценено как $\Theta(\log_2 n)$.

1.3. Свойства дискретных паттернов для прямых на плоскости

Известно, что через любые две различные точки на плоскости всегда можно провести прямую и притом только одну. То же верно для любой гиперплоскости при числе точек общего положения, равном размерности пространства. Интересно проверить, сохраняется ли данное свойство при переходе от непрерывных прямых к дискретным паттернам, в случае данной работы – к ближайшему и диадическому.

Очевидно, что для диадических паттернов верно, что на изображении (с линейным размером больше двух) всегда найдутся два различных пикселя, через которые проходит более одного паттерна. Примером может служить пара соседних пикселей на изображении. Остается вопрос: возможно ли через любые два пикселя изображения провести дискретный паттерн?

Здесь и далее *системой паттернов* будем называть набор паттернов, заданных для изображения размера n. *Ближайшей системой* будем называть набор ближайших паттернов, аппроксимирующих геометрические прямые, заданные с помощью (s, t)-параметризации, где s, t – целые числа и $t \in [0; n-1]$ и $s \in [0; n-1]$. *Системой БПХ* будем называть набор диадических паттернов, заданных аналогично.

Теорема 1. *Для любых двух пикселей на изображении со стороной $n = 2^m$, где m–натуральное число, существует проходящий через них диадический паттерн системы БПХ.*

Доказательство. Рассмотрим часть паттерна P, заключенную между абсциссами $x_2 > x_1$. Символом t_p обозначим наклон такого подпаттерна. Ясно, что если $x_2 - x_1 = n - 1$, то утверждение верно по определению системы паттернов БПХ. Заметим, что утверждение верно для любой пары $x_2 > x_1$, если t_p принимает все значения в диапазоне $[0, x_2 - x_1]$. Действительно, $0 \le y_2 - y_1 \le x_2 - x_1$, поскольку мы рассматриваем ПГП с наклоном вверх. А, если существует паттерн с наклоном подпаттерна $t_p = y_2 - y_1$, то при некотором s концы подпаттерна будут иметь координаты (x_1, y_1) и (x_2, y_2) соответственно.

Докажем, что t_p принимает все значения в диапазоне $[0, x_2 - x_1]$. Ясно, что граничные подпаттерны, а именно горизонтальный $t_p^{min} = 0$ и диагональный $t_p^{max} = x_2 - x_1$, существуют, поскольку паттерны с $t = 0$ и $t = n - 1$ принадлежат системе.

Теперь рассмотрим зависимость $t_p(t)$: если $\Delta t_p(t) = t_p(t+1) - t_p(t) \leq 1$, то t_p будет пробегать все значения в диапазоне $[t_p^{min}, t_p^{max}]$ (в силу существования граничных паттернов), а значит, для любой пары рассматриваемых пикселей найдется паттерн, проходящий через них.

Для того, чтобы доказать, что $\Delta t_p(t)$ действительно не превышает единицы, рассмотрим смещение каждого пикселя паттерна при увеличении наклона $\Delta D_t^n(x) = D_{t+1}^n(x) - D_t^n(x)$ и покажем, что $0 \leq \Delta D_t^n(x) \leq 1$. Воспользуемся методом математической индукции: для диадического паттерна размером $n = 2^1$ это неравенство верно, тогда пусть для $n = 2^k$ оно также верно, докажем его для $n = 2^{k+1}$.

Рассмотрим два случая приращения наклона t на единицу: четный и нечетный. Пусть t – четный, в этом случае паттерн с наклоном $t+1$ отличается от предыдущего тем, что ординаты всех пикселей его правой половины увеличены на единицу. Следовательно, для любого его подпаттерна $0 \leq \Delta D_t^n(x) \leq 1$, что следует из определения (4).

Если t – нечетный, при инкременте наклона t становится четным, и, как следует из определения (4), сдвиг в центре паттерна исчезает, однако прибавляется по одной единице сдвига на каждую (в силу симметрии) половину паттерна. Поскольку (из предположения индукции) для обеих половин длины $n = 2^k$ изменение наклона каждого подпаттерна удовлетворяет ограничению $0 \leq \Delta D_t^n(x) \leq 1$, и эти половины не сдвинуты друг относительно друга, то любой подпаттерн всего паттерна удовлетворяет данному ограничению. То есть $0 \leq \Delta D_t^n(x) \leq 1$, а, поскольку $\Delta t_p(t) = \Delta D_t^n(x_2) - \Delta D_t^n(x_1)$, то $-1 \leq \Delta t_p(t) \leq 1$.

Таким образом, для любых x_1, x_2 величина $t_p(t)$ пробегает все значения в диапазоне $(0, x_2 - x_1)$, а, следовательно, через любую пару пикселей можно провести диадический паттерн. ■

Будем называть такое свойство дискретных паттернов *полнотой*. Важно, что для системы БПХ нельзя удалить ни одного паттерна так, чтобы свойство полноты не нарушилось. Для того, чтобы убедиться в этом, достаточно рассмотреть крайние пиксели удаленного паттерна.

Система ближайших паттернов также полна. Доказательство строится аналогично теореме 1, однако то, что при увеличении t на единицу приращение ординат пикселей равно 0 или 1, следует из определения ближайшего паттерна.

2. ТРЕХМЕРНОЕ БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ХАФА ДЛЯ ПЛОСКОСТЕЙ

В трехмерном случае термин “диадический паттерн” может использоваться как для обозначения аппроксимации прямой, так и плоскости. Во избежание путаницы, в первом случае будем использовать термин “диадическая прямая”, а во втором – “диадическая плоскость”.

2.1. Параметризация и паттерны дискретных плоскостей

Введем типизацию плоскостей и способ их задания. Все плоскости в трехмерном пространстве можно разделить на три группы, атрибутируя их соответствующей координатной осью по следующему принципу: будем называть плоскость преимущественно-перпендикулярной некоторой координатной оси, если наименьший угол между нормалью к плоскости и этой осью меньше или равен углу, образованному с другими осями. Далее, без ограничения общности будем рассматривать только одну из групп, а именно – преимущественно-перпендикулярную оси x, смотри рисунок 2.

Для трехмерного случая аналогично вводится (s,t)-параметризация, где s – параметр сдвига, а $t = (t_1, t_2)$ – вектор наклона, таким образом тремя точками $(s, 0, 0)$, $(s + t_1, 0, n-1)$, $(s + t_1 + t_2, n-1, n-1)$ на рёбрах куба (центры пикселей трехмерного изображения) задаётся плоскость.

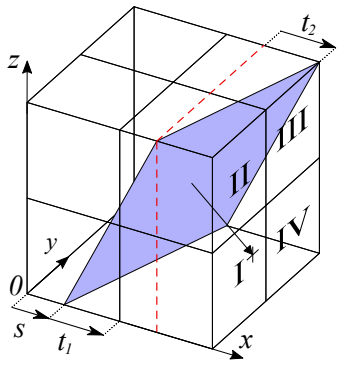

Рис. 2. Иллюстрация (s,t)-параметризации плоскости в трехмерном пространстве.

Преимущественно-перпендикулярные оси x плоскости, в свою очередь, подразделяются знаками t еще на четыре подгруппы так, как показано на рисунке 2. Таким образом, мы выделили 12 подтипов плоскостей. Далее будем рассматривать преимущественно-перпендикулярные оси x плоскости.

Рассмотрим структуру и способ представления координат пикселей диадической плоскости. Аналогично двумерному случаю паттерн диадической плоскости определяется параметрами наклона (t_1, t_2), а сдвиг s, в свою очередь, определяет его положение, как показано на рисунке 3.

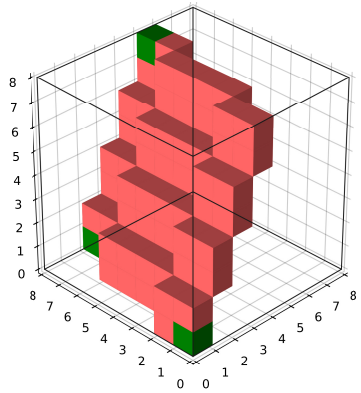

Рис. 3. Пример структуры диадической плоскости. Зеленым цветом отмечены воксели, задающие плоскость.

Формула определения абсциссы центров пикселей плоскости, проходящей через точку $(0,0,0)$ (см. рисунок 2) с параметрами t_1 и t_2, записывается следующим образом:

$$x = D^n_{t_1,t_2}(y,z) = D^n_{t_1}(y) + D^n_{t_2}(z), \tag{7}$$

а её обобщение на m-мерный случай – так:

$$x_0 = D^n_t(\vec{x}) = \sum_{i=1}^{m-1} D^n_{t_i}(x_i), \tag{8}$$

где $\vec{x} = (x_1, ..., x_{m-1})$.

Ортотропная ошибка аппроксимации диадической плоскости может быть вычислена, как сумма ошибок её образующих, являющихся диадическими паттернами (что следует из (7)), поскольку дли них эта ошибка измеряется в направлении общей оси x. Таким образом, для равностороннего изображения с линейным размером n в худшем случае (четное m) диадическая

плоскость с параметрами $s = 0, t_1 = \frac{n-1}{3}, t_2 = \frac{n-1}{3}$ будет иметь максимальную ортотропную ошибку, равную $\frac{\log_2 n}{3}$.

Заметим, что ортогональная $N^n_{t_1,t_2}(x)$ и ортотропная $E^n_{t_1,t_2}(x)$ ошибки для плоскостей связаны аналогично двумерному случаю:

$$N^n_{t_1,t_2}(x) = \frac{E^n_{t_1,t_2}(x)}{\sqrt{1 + (\frac{t_1}{n-1})^2 + (\frac{t_2}{n-1})^2}}. \tag{9}$$

Таким образом, максимальная ортогональная ошибка аппроксимации ограничена сверху величиной $\frac{\log_2 n}{3}$, а снизу – $\frac{\log_2 n}{\sqrt{11}}$ (для плоскости с вектором наклонов $((n-1)/3, (n-1)/3)$).

Не смотря на результат, полученный в разделе 1.3, утверждение, что через любые три вокселя проходит диадическая плоскость, то есть ТБПХ для плоскостей обладает свойством *полноты*, вовсе не очевидно, и этот вопрос не будет рассмотрен в данной работе.

2.2. Алгоритм трехмерного быстрого преобразования Хафа для плоскостей

Впервые алгоритм ТБПХ для плоскостей был предложен в работе [18] в 2016 году. В данной главе предлагается более подробное описание собственно алгоритма и его свойств. Процедура вычисления БПХ для плоскостей приведена на языке MATLAB ниже (смотри алгоритм 2).

```
function h = fht3plane(m)
    sz = size(m);
    n = sz(1);

    hs = zeros(sz);
    for i = 1 : n
        hs(:, :, i, :) = fht2(m(:, :, i, :));
    end

    h = zeros(sz);
    for i = 1 : n
        h(:, i, :, :) = fht2(hs(:, i, :, :));
    end
end
```

Листинг 2. Алгоритм трехмерного быстрого преобразования Хафа для плоскостей преимущественно-перпендикулярных оси x.

Входом алгоритма является трехмерное изображение, а выходом – соответствующий Хаф-образ. На первом этапе вычисляется преобразование каждого слоя (двумерного массива) данных с координатой $z = z_i$ исходного изображения m, на основе чего формируется промежуточное трехмерное изображение hs, в каждый слой которого с аппликатой z_i записывается результат вычисления БПХ для соответствующего слоя исходного массива. Заметим, что суммы по множествам вокселей вида $\Omega(x_i, y_i) = \{(x_i, y_i, z) | z \in [0, .., n-1]\})$ в hs эквивалентны суммам по вокселям диадических плоскостей в исходном пространстве с параметрами $s = x_i, t_1 = y_i, t_2 = 0$. Этот пример иллюстрирует интересное свойство: сумма по вертикальным прямым в кубе hs эквивалентна в исходном пространстве суммированию по диадическим плоскостям с нулевым t_2. Далее, чтобы подсчитать суммы по всем преимущественно-перпендикулярным оси x плоскостям одного типа, достаточно вычислить, аналогично первому этапу, БПХ для каждой плоскости, перпендикулярной y. Именно это и делается в приведенном алгоритме. При этом строится Хаф-образ только для одного из 12 подтипов, остальные подтипы могут быть получены независимым изменением направления циклической перестановки в двух итерациях подсчета послойных БПХ и сменой ведущей координаты.

Нетрудно оценить вычислительную сложность такого алгоритма: для $2n$ плоскостей выполняется расчет БПХ, сложность которого составляет $n^2 \log_2 n$. Таким образом, при расчете всех подтипов диадических плоскостей, результирующая вычислительная сложность алгоритма последовательного подсчета БПХ составляет $24n^3 \log_2 n$. Впрочем, этот результат может быть улучшен. Заметим, что результат для каждого из 12 подтипов может быть получен двумя способами: первый и второй проход БПХ в алгоритме перестановочны. Теперь рассмотрим пару подтипов, области классификации которых (отмечены римскими цифрами на рис. 2) смежные, но принадлежат разным граням. Они различаются только одним из двух типов БПХ в отдельных проходах. Значит, вместе их можно вычислить за 3, а не за 4 прохода, сохранив однажды вычисленный промежуточный массив. Это и позволяет уменьшить вычислительную сложность до $18n^3 \log_2 n$ операций.

Для случая параллельных вычислений сложность будет составлять $2 \log_2 n$, но с использованием уже $12n^3$ последовательных вычислителей.

3. ТРЕХМЕРНОЕ БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ХАФА ДЛЯ ПРЯМЫХ

В данном разделе впервые предлагается алгоритм быстрого трехмерного преобразования Хафа для прямых, соответствующий преобразованию Йона для непрерывного случая [22]. Алгоритм имеет не улучшаемую асимптотически сложность последовательного вычисления.

3.1. Параметризация и паттерн для дискретных прямых в пространстве

Следуя уже предложенной в предыдущих разделах схеме, рассмотрим (s,t)-параметризацию прямой в трехмерном пространстве. Разделим прямые на три типа, атрибутируя их соответствующей координатной осью, к примеру, будем называть прямую преимущественно ориентированной вдоль некоторой оси, если острый угол между ней и данной прямой меньше или равен углу между данной прямой и другими осями. Здесь и далее без ограничения общности будем рассматривать преимущественно ориентированной вдоль оси z прямые. Для определения такой прямой нам достаточно задать вектор (s_1, s_2, t_1, t_2), который в свою очередь задаёт две точки: $(s_1, s_2, 0)$ и $(s_1 + t_1, s_2 + t_2, n-1)$, через которые проходит прямая (смотри рисунок 4).

Внутри одного типа будем подразделять линии на 4 подтипа, в зависимости от знаков параметров t_1 и t_2 (знак определяется направленностью относительно координатных осей x и y соответственно). В работе будем рассматривать случай, когда оба знака положительны (соответствует подтипу III на рисунке 4).

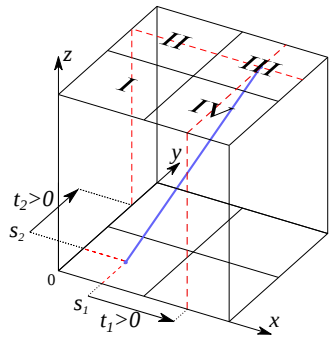

Рис. 4. Иллюстрация параметризации преимущественно ориентированной вдоль оси z прямой в трехмерном пространстве.

На рисунке 5 представлен пример структуры диадической прямой в трехмерном случае. Заметим, что её проекции на плоскости $0xz$ и $0yz$ являются плоскими диадическими паттернами, что следует из устройства алгоритма, который будет изложен далее.

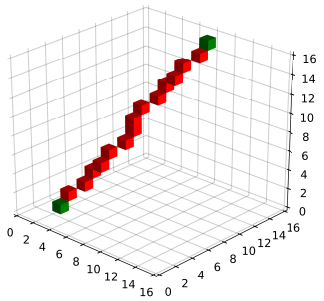

Рис. 5. Иллюстрация структуры диадической прямой в трехмерном изображении. Зеленым отмечены воксели, задающие диадическую прямую.

Выпишем систему уравнений для определения координаты центра вокселя (x, y, z), принадлежащего диадической прямой, аналогично формуле (5):

$$\begin{cases} x(z) = D^n_{t_1}(z) + s_1 = \sum\limits_{k=0}^{p-1} t_{1k} \left[\frac{2^k z}{n-1} \right] + s_1, \\ y(z) = D^n_{t_2}(z) + s_2 = \sum\limits_{k=0}^{p-1} t_{2k} \left[\frac{2^k z}{n-1} \right] + s_2, \end{cases} \tag{10}$$

где t_{1k}, t_{2k} – значения k-го бита в числах t_1 и t_2 соответственно.

Два вокселя на трехмерном изображении, как и в плоском случае, могут задавать не единственную диадическую прямую, но для любых двух вокселей на изображении всегда найдется хотя бы одна диадическая прямая, через них проходящая. Рассмотрим два вокселя с координатами $V_1(x_1, y_1, z_1)$ и $V_2(x_2, y_2, z_2)$. В разделе 1.3 было доказано, любые два пикселя на плоском изображении определяют хотя бы один диадический паттерн. Таким образом для проекций вокселей на плоскости $0xz$ и $0yz$ верно, что существуют два диадических паттерна, проходящих через пары точек $(x_1, 0, z_1)$, $(x_2, 0, z_2)$ и $(0, y_2, z_2)$, $(0, y_2, z_2)$ соответственно. Эти паттерны, в свою очередь, задают в свою очередь трехмерную диадическую прямую по формуле (10), проходящую через воксели V_1 и V_2. Резюмируя, поскольку мы всегда можем построить проекции диадической прямой, мы можем построить и саму прямую.

Оценим ортогональную $N^n_{t_1,t_2}(x)$ и ортотропную $E^n_{t_1,t_2}(x)$ ошибки аппроксимации диадической прямой. Используя, формулу (6) и факт того, что величина ортотропной ошибки диадического паттерна на плоскости равна $\frac{1}{6}\log_2 n$, можно получить аналогичную оценку для трехмерной диадической прямой. Максимальная ортотропная ошибка аппроксимации равна $\frac{\sqrt{2}}{6}\log_2 n$: поскольку ортотропные ошибки в плоскостях Oyz и Oxz независимы, а плоскости – перпендикулярны, необходимо вычислить длину вектора, равного сумме двух векторов ортотропных ошибок, ортогональных друг другу.

Аналогично предыдущим разделам, получим зависимость между ортогональной и ортотропной ошибками аппроксимации. Поскольку угол между ними равен углу между рассматриваемой диадической прямой и координатной плоскостью $0yz$, то нетрудно получить, что ошибки связаны выражением:

$$N^n_{t_1,t_2}(x) = \frac{E^n_{t_1,t_2}(x)}{\sqrt{1 + (\frac{t_1}{n-1})^2 + (\frac{t_2}{n-1})^2}}. \tag{11}$$

Верхняя оценка ортогональной ошибки аппроксимации, как и в предыдущих случаях определяется верхней оценкой ортотропной, $\frac{\sqrt{2}}{6}\log_2 n$, а достижимая ошибка по данной формуле, для наклонов $t_1 = t_2 = (n-1)/3$ (когда достигается пиковая ошибка каждой из проекций диадической прямой), составляет $\frac{\log_2 n}{\sqrt{22}}$.

3.2. Алгоритм вычисления быстрого трехмерного преобразования Хафа для прямых

Рассмотрим алгоритм вычисления трехмерного быстрого преобразования Хафа для прямых. Код данного алгоритма приведен ниже на языке MATLAB (смотри алгоритм 3). Основной функцией алгоритма является $fht3line$, входом которой является четырехмерное многоканальное изображение m, а выходом – Хаф-образ для него.

```
function h = fht3line(m)
  n = size(m, 3);
  if n < 2
    h = m;
    return
  end

  n0 = round(n / 2);
  h = mergeHT(fht3line(m(:, :, 1 : n0, :, :)), fht3line(m(:, :, n0 + 1 : n, :, :)));
end

function h = mergeHT(h0, h1)
  m = size(h0, 1);
  n0 = size(h0, 3);
  o = size(h0, 5);

  n = 2 * n0;
  h = zeros(m, m, n, n, o);
  r = (n0 - 1) / (n - 1);

  for i = 1 : n
    for j = 1 : n
      t = [i - 1, j - 1];
      t0 = round(t .* r);
      s = t - t0;

      h(:, :, i, j, :) = h0(:, :, t0(1) + 1, t0(2) + 1, :) + ...
                         [[
                            h1(s(1) + 1 : m, s(2) + 1 : m, t0(1) + 1, t0(2) + 1, :), ...
                            h1(s(1) + 1 : m, 1 : s(2), t0(1) + 1, t0(2) + 1, :)
                          ];
                          [
                            h1(1 : s(1), s(2) + 1 : m, t0(1) + 1, t0(2) + 1, :), ...
                            h1(1 : s(1), 1 : s(2), t0(1) + 1, t0(2) + 1, :)
                         ]];
    end
  end
end
```

Листинг 3. Алгоритм быстрого преобразования Хафа для одного из 12 подтипов прямых.

Остановимся на вопросе “Почему входное изображение четырехмерное?”. Ответ кроется в рекурсивной реализации алгоритма: размерность входа должна совпадать с размерностью выхода, а, поскольку пространство Хафа для данного алгоритма четырехмерно (что определяется размерностью вектора (s,t)-параметризации), то и входное изображение должно быть

четырехмерным, поэтому исходное трехмерное изображение дополняется еще одним измерением, таким образом, что ему присваивается единичная координата по четвертой компоненте, остальным элементам массива присваивается нулевое значение.

В строке 27 алгоритма 3 написана операция поэлементной сумации двумерных сечений массива с определенным сдвигом. Интересно заметить, что структура алгоритма совпадает со структурой БПХ (алгоритм 1), за тем лишь исключением, что суммируются двумерные массивы, а не одномерные. Отсюда также видно, что проекция диадической прямой на плоскость $0xz$ или $0zy$ является в этом случае диадическим паттерном.

Оценим сложность вычисления предложенного алгоритма. На первом шаге выполняется суммирование по всем парам плоскостей четырьмя способами, то есть $4n^2 \cdot n/2 = 2n^3$ операций. На следующем шаге выполняется суммирование по всем парам изображений (по *слоям*) высоты два (всего таких пар $n/4$), притом для каждого из четырех вариантов наклона подпаттернов высоты два выполняется 4 суммирования. В каждом слое число паттернов одного типа равно n^2, то есть на данном этапе выполняется $4 \cdot 4n^2 \cdot n/4 = 4n^3$ операций, на следующем уровне – $8n^3$ и число таких слагаемых равно глубине рекурсии алгоритма, то есть $\log_2 n$. Нетрудно посчитать, что итоговая вычислительная сложность алгоритма быстрого трехмерного преобразования Хафа составляет $2n^3+4n^3+16n^3+...+n^4 = 2n^3(n-1)$ операций. Суммарная сложность по всем типам прямых составляет $12n^3(n-1)$ операций. То есть асимптотическая вычислительная сложность алгоритма равна $\Theta(n^4)$ операций. Поскольку требуется заполнить $\Theta(n^4)$ значений ответа, то невозможно создать быструю вычислительную схему подсчета быстрого трехмерного преобразования Хафа для прямых, превосходящую данную по асимптотической сложности. Имея же $\Theta(n^4)$ процессоров можно вычислить ТБПХ для прямых за $\Theta(\log_2 n)$ операций.

4. ЗАКЛЮЧЕНИЕ

В работе предложен новый алгоритм трехмерного быстрого преобразования Хафа для прямых, показано, что асимптотическая оценка его сложности составляет $\Theta(n^4)$ операций. Таким образом показано, что ТБПХ для прямых является асимптотически неулучшаемым среди алгоритмов суммации по дискретным прямым. Для всех рассматриваемых в работе алгоритмов быстрого вычисления преобразования Хафа изучены свойства соответствующих им диадических паттернов, в том числе полнота и точность аппроксимации их геометрического прообраза.

СПИСОК ЛИТЕРАТУРЫ

1. Hough P. Machine analysis of bubble chamber pictures. *International conference on high energy accelerators and instrumentation*, 1959, no. 10, p. 2.

2. Mukhopadhyay P., Chaudhuri B. A survey of Hough Transform. *Pattern Recognition*, 2015, vol. 48 no. 3, pp. 993-1010.

3. Shehata A., Mohammad S., Sameer M., Rahab E. A survey on Hough transform, theory, techniques and applications, *arXiv preprint arXiv:1502.02160.*

4. Xu L., Oja E., Kultanan P., A new curve detection method: Randomized Hough transform (RHT), *Pattern Recognitoin*, 1990, no. 11, pp. 331-338.

5. Fischler M.A., Bolles R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM.*, vol. 24, no. 6, pp. 381–395.

6. Jiri M., Galambos C., Kittler J. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 2000, vol. 78, no. 1, pp. 119-137.

7. Limberger, F.A., Oliveira, M.M. Real-Time Detection of Planar Regions in Unorganized Point Clouds. *Pattern Recognition*, 2014, vol. 48, no. 6, pp. 2043–2053.

8. Brady, Martin L., and Whanki Yong. Fast parallel discrete approximation algorithms for the Radon transform. *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*. ACM, 1992.

9. Gotz W. A. Eine Schnelle Diskrete Radon Transformation basierend auf rekursiv definiertern. *Digitalen Geraden*, 1993.

10. Vuillemin J. E. Fast linear Hough transform. Application Specific Array Processors. *International Conference on IEEE*, 1994, pp. 91–99.

11. Карпенко С. М., Николаев Д.П., Николаев П.П, Постников В. В. Быстрое преобразование Хафа с управляемой робастностью. *Искусственные интеллектуальные системы и Интеллектуальные САПР. Труды международной конференции IEEE AIS'04 и CAD-2004.*. Москва: Изд-во Физматлит, 2004, т. 2, стр. 303-309.

12. Frederick M. T., VanderHorn N. A., Somani A. K. Real-time H/W implementation of the approximate discrete Radon transform. *16th IEEE International Conference on IEEE*, 2005, pp. 399-404.

13. Арлазаров, В.В., Жуковский, А.Е., Кривцов, В.Е., Николаев, Д.П. and Полевой, Д.В., Анализ особенностей использования стационарных и мобильных малоразмерных цифровых видео камер для распознавания документов. *Информационные технологии и вычислительные системы*, 2014, vol. 3, pp.71-81.

14. D. Krokhina, V. Blinov, S. Gladilin, I. Tarhanov, V. Postnikov. Fast roadway detection using car cabin video camera. *Eighth International Conference on Machine Vision (ICMV 2015)*. Proceedings SPIE, 2015, V. 9875, pp. 98751F1-5.

15. Nikolaev D.P., Karpenko S.M., Nikolaev I.P., Nikolayev P.P. Hough transform: underestimated tool in the computer vision field. *Proceedings of the 22th European Conference on Modelling and Simulation*, 2008, pp. 238-246.

16. Прун В.Е., Бузмаков А.В., Николаев Д.П., Чукалина М.В., Асадчиков В.Е. Вычислительно эффективный вариант алгебраического метода компьютерной томографии. *Автоматика и телемеханика*, 2013, №10, стр. 86–97.

17. Кунина И.А., Гладилин С.А., Николаев Д.П. Слепая компенсация радиальной дисторсии на одиночном изображении с использованием быстрого преобразования Хафа. *Компьютерная оптика*, 2016, том. 40, № 3, стр. 395-403.

18. 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.

19. Роджерс Д. Алгоритмические основы машинной графики. *Москва: Мир*, 1989. ISBN 5-03-000476-9.

20. Ershov E.I., Shvets E.A., Khanipov T.M., Nikolaev D.P. Generation algorithms of fast generalized hough transform. *Proceedings 30th European Conference on Modelling and Simulation*, 2017, (in print).

21. Ershov, E., A. Terekhin, D. Nikolaev, V. Postnikov, and S. Karpenko. Fast Hough transform analysis: pattern deviation from line segment. *In Eighth International Conference on Machine Vision*, International Society for Optics and Photonics, 2015, pp. 987509-987509., 2015.

22. Гельфанд, И.М., Гиндикин, С.Г., Граев, М.И. Избранные задачи интегральной геометрии. *Москва: Добросвет*, 2000.

23. Безматерных П.В., Вылегжанин Д.В., Гладилин С.А., Николаев Д.П. Генеративное распознавание двумерных штрихкодов. *Искусственный интеллект и принятие решений*, 2010. № 4. стр. 63-69.

24. Ershov E.I., Karpenko S.M. Fast Hough Transform and approximation properties of dyadic patterns, *arXiv:2096256*.

Fast Hough transform generalization for three-dimensional images

Ershov E.I., Terekhin A.P. and Nikolaev D.P.

Abstract—The paper is devoted to the algorithms investigation for calculating the fast Hough transform for two-dimensional and three-dimensional images. A method is proposed for calculating the fast Hough transform for straight lines in a three-dimensional image. The asymptotic complexity and required memory of the proposed method are $O(n^4)$, where n is the linear size of the original image. The algorithms of FHT for approximation in two-dimensional and three-dimensional spaces are also considered. The properties of accuracy and completeness of the corresponding sets of dyadic patterns are investigated.

.

KEYWORDS: descrete Radon transform, descrete John transform, three-dimensional Hough transform, fast Hough transform, Hough transform precision, Hough transform complexity