

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК БЕЛАРУСИ
ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ

А.О. Макаров, В.В. Старовойтов

**БЫСТРЫЕ АЛГОРИТМЫ
ВЫЧИСЛЕНИЯ ПРИЗНАКОВ
НА ЦИФРОВЫХ ИЗОБРАЖЕНИЯХ**

Препринт № 1

Минск 2005

УДК 681.3.01.+681.327.12

Макаров А.О., Старовойтов В.В. **Быстрые алгоритмы вычисления признаков на цифровых изображениях.** – Минск, 2005. – 39 с. (Препринт / Объединенный институт проблем информатики Национальной академии наук Беларуси; № 1).

Рассмотрены быстрые алгоритмы вычисления признаков. Выполнен анализ алгоритмов ускорения дискретной свертки, рекурсивных алгоритмов фильтрации и алгоритмов обнаружения объектов, а также усовершенствованы наборы фильтров Габора. Даны описания алгоритмов и рекомендации по их использованию. Представлены результаты функционирования алгоритмов.

Ил. 19, табл. 1, библи. 59 назв.

Рецензент

кандидат физико-математических наук В.А. Мильман

© Макаров А.О., Старовойтов В.В.,
2005

Введение

Одной из актуальных задач в области распознавания образов является задача вычисления признаков объектов по их изображениям. Признаки объекта – это его качественные и количественные характеристики. Исследуемый объект описывается на языке выбранных признаков на основе априорных данных. Отбор признаков тесно связан с решаемой задачей. Например, изображение одного и того же объекта может породить разные множества признаков, а одни и те же признаки могут описывать объекты разных классов.

В разных задачах один и тот же объект может описываться разными наборами признаков. При этом для уменьшения избыточности информации желательно минимизировать число признаков без существенного увеличения вероятности ошибки распознавания, исключить признаки, зависящие от других, обеспечить возможность использования относительно простых алгоритмов распознавания [1]. Сама по себе задача отбора признаков в общем виде не может быть формализована.

В некоторых случаях допускается некоторая избыточность признаков. Исходное множество признаков набирается эмпирическим путем, затем оно по возможности уменьшается посредством их анализа. Снижение размерности признакового пространства можно выполнить различными способами [1–3]. При этом желательно отобрать такие признаки, которые наиболее инвариантны к изменениям визуального представления объекта и оптимальным образом описывают исследуемые объекты. Под инвариантностью понимается независимость от условий получения изображений и изменений (яркостных и аффинных) исследуемых объектов, представленных на изображениях.

Изображение содержит объекты, характеризующиеся свойственными только им признаками. Необходимо вычислить признаки этих объектов. Набор признаков используется для описания модели объекта и его распознавания. Например, для распознавания геометрических фигур используются их формы. При распознавании лица человека используются наиболее подходящие признаки, т. е. такие, которые наилучшим, минимальным и наименее избыточным способом определяют и характеризуют данный объект (рис. 1).

Для многих прикладных систем распознавания важна скорость вычисления признаков. Обзору алгоритмов, предназначенных для быстрого вычисления признаков, посвящена данная работа. Качественные признаки объектов (например, больше–меньше, лучше–хуже) в данной работе не рассматриваются.

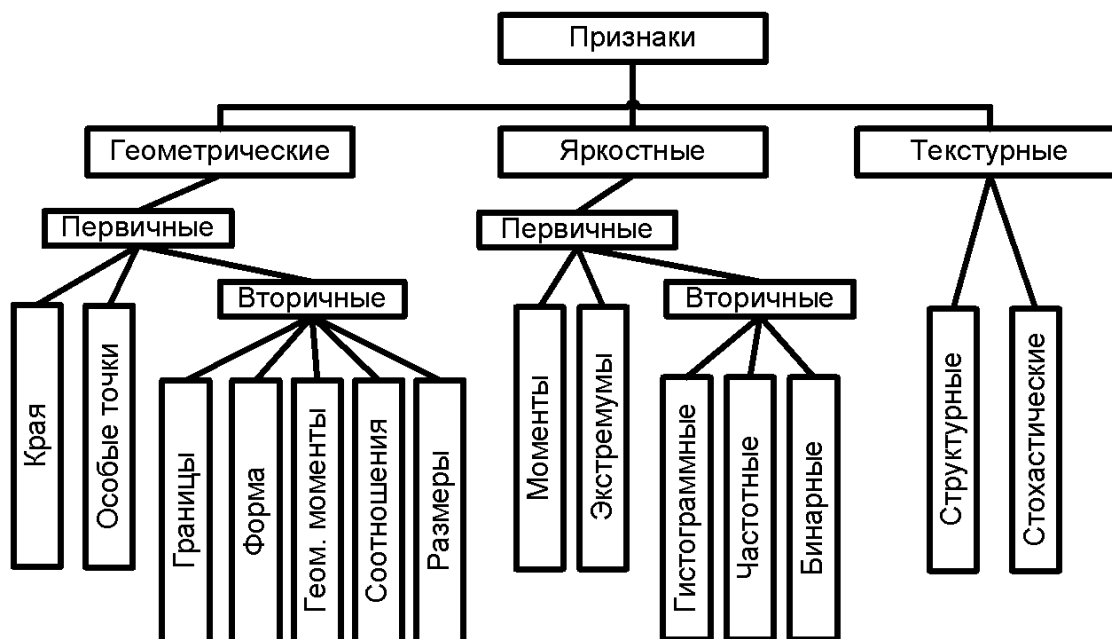


Рис. 1. Классификация признаков разных типов

1 Признаки объектов, вычисляемые на цифровых изображениях

Объект характеризуется некоторыми глобальными и локальными признаками. Количество признаков объекта определяет размерность признакового пространства, в котором объект определен. Глобальные признаки отличают объект от всех других объектов, находящихся на сцене. Локальные признаки определяют местоположение объекта на сцене.

Наборы признаков описываются в виде вектора, который является средством компактного и удобного представления изображений, их частей или отдельных объектов. Вектор содержит измерения, которые могут быть количественными или качественными. Количественные признаки характеризуются числовыми значениями. Качественные признаки определяются человеком и зачастую носят субъективный характер.

Признаки объектов на цифровых изображениях можно разделить на первичные и вторичные. Первичные признаки – это такие признаки, которые вычисляются с помощью цифрового изображения. Вторичные признаки, как правило, вычисляются на базе первичных.

1.1 Геометрические признаки

Геометрические признаки вычисляются локально, при этом предполагается, что изображение объекта достаточно однородно и позволяет увидеть и вычислить первичные геометрические признаки.

Например, к первичным геометрическим признакам можно отнести:

- край объекта, участок линии (они характеризуются заметным перепадом яркости или цвета);
- угол;
- Т-, Y-, L-образные соединения линий;
- начало, конец линии;
- пятно (область);
- выпуклость (вогнутость) объекта.

К вторичным признакам можно отнести:

- площадь;
- расстояние между заданными точками;
- периметр.

1.1.1 Особые точки

Объекты могут иметь свойственные им геометрические особые точки. Например, особыми точками прямоугольника являются четыре его вершины, а отрезка – два конца отрезка. Одними из особых точек объектов на изображении являются крайние точки фигуры объекта. Крайние точки объекта: вершины углов фигуры этого объекта; самая верхняя и самая нижняя точки фигуры объекта относительно координатных осей; левая нижняя, левая верхняя, правая нижняя, правая верхняя точки фигуры объекта; концы линий. Особые точки используются для вычисления вторичных признаков.

1.1.2 Границы объекта

Края объекта формируют его границы. Границы объекта являются связными контурными линиями объекта, которые характеризуют его форму [4]. Данный признак используется для вычисления таких геометрических признаков, как размер, форма и ориентация объекта. Границы являются вторичными признаками.

1.1.3 Форма объекта

Форма объекта определяется контуром и границами объекта [4], характеризуется наличием дыр, отношением площади к периметру, вытянутостью, округлостью объекта. Геометрические признаки, определяющие форму объекта, состоят из периметра, площади, минимального и максимального радиусов, углов, степени округленности объекта, кривизны объекта, степени симметрии объекта. Форма является вторичным признаком.

1.1.4 Размеры объекта

К размерным признакам можно отнести количество пикселей объекта, его длину, ширину, габариты [5–6].

1.2 Яркостные признаки

Использование данных признаков применимо к цветным, полутоновым и бинарным изображениям. Данные признаки описывают изменения локальной энергии изображения – перепады в яркостях пикселей, составляющих изображение. Например, если на изображении присутствует относительно равномерный фон, а на фоне располагается объект с отличающейся от фона яркостью, то в данном случае границы объекта формируются яркостными признаками, так как имеются перепады в уровнях яркостей фона и объекта.

1.2.1 Экстремумы

Экстремум – это максимальное или минимальное значение пикселя или пикселей на изображении. Если он вычисляется на всем изображении, то это глобальный экстремум, если на области изображения – то локальный. Экстремумы являются первичными признаками.

1.2.2 Частотные признаки

Некоторые преобразования (например, Фурье), позволяют сформировать частотное представление изображения, по которому можно вычислить частотные признаки. Высокочастотные признаки используются для обнаружения краев и границ, угловые щели используются для определения ориентации. Например, изображение, содержащее несколько параллельных линий с углом ориентации θ , даст четкую энергию в частотном пространстве вдоль линии с углом $\frac{\pi}{2} + \theta$, проходящей через начало координат преобразования Фурье. Поскольку сначала вычисляется частотное представление изображения, а на частотном пространстве – частотные признаки, то они являются вторичными.

1.2.3 Гистограммные признаки

Гистограммные признаки являются статистическими и описывают распределение яркости изображения. К этому типу признаков относят среднее значение, стандартное отклонение, искажение, энергию, энтропию. Гистограммные признаки являются вторичными.

1.2.4 Моменты

Рассмотрим неотрицательную функцию $f(x, y)$, нормализованную так, что заключенный под функцией объем равен единице. Момент m_{pq} функции двух переменных $f(x, y)$ определяется следующим образом:

$$m_{pq} = \int \int x^p y^q f(x, y) dx dy. \quad (1)$$

Здесь $p, q = 0, 1, 2, \dots$, где p, q – неотрицательные целые числа, а $(p + q)$ – порядок момента.

Пусть $g(i, j)$ описывает некоторое полутонное изображение размером $m \times n$. Если на нем представлен один объект, то моменты позволяют построить описание объекта, на их основе вычисляются признаки, инвариантные к аффинным преобразованиям объекта.

Формула моментов порядка p, q изображения g размером $m \times n$ записывается следующим образом:

$$m_{pq} = \sum_{i=1}^m \sum_{j=1}^n i^p j^q g(i, j), \quad (2)$$

где i, j – координаты пикселя с яркостью $g(i, j)$.

Центральные моменты порядка p, q изображения g размером $m \times n$ записываются следующим образом:

$$\mu_{pq} = \sum_{i=1}^m \sum_{j=1}^n (i - \bar{i})^p (j - \bar{j})^q g(i, j), \quad (3)$$

где $\bar{i} = \frac{m_{10}}{m_{00}}$ и $\bar{j} = \frac{m_{01}}{m_{00}}$ – координаты центра тяжести изображения.

Моменты являются первичными признаками.

1.3 Текстурные признаки

Текстуры встречаются на самых разных изображениях, например на изображениях песка, зерна, травы, ткани. Термин «текстура» связан с повторением базовых элементов текстур – текселей. Тексель содержит несколько пикселей, расположение которых периодичное, квазипериодичное или случайное [4]. Природные текстуры обычно случайные, в то время как искусственные очень часто периодические. Текстуры могут быть грубыми, тонкими, размытыми, гранулированными, регулярными, нерегулярными, линейными. Текстуры обычно делятся на структурные и стохастические [4]. В работе [7] представлен набор из 28 текстурных признаков. Далее представлены текстурные признаки из этого набора:

– угловой момент второго порядка

$$f_1 = \sum_i \sum_j \{p(i, j)\}^2; \quad (4)$$

– контраст

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right\}; \quad (5)$$

– корреляция

$$f_3 = \frac{\sum_i \sum_j (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}; \quad (6)$$

– сумма дисперсии

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i); \quad (7)$$

– сумма энтропии

$$f_8 = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}; \quad (8)$$

– энтропия

$$f_9 = - \sum_i \sum_j p(i, j) \log(p(i, j)). \quad (9)$$

Здесь $p(i, j)$ – (i, j) -й элемент матрицы встречаемости яркостей (МВЯ); $\mu_x, \mu_y, \sigma_x, \sigma_y$ – средние значения и стандартные отклонения p_x и p_y ; $p(i)$ – i -й элемент матрицы предельной вероятности, полученной путем суммирования строк матрицы $p(i, j)$; N_g – количество уровней серого в изображении.

Для изображения вычисляются МВЯ $CM_{lm} = \{P_{ij}\}$ размером $G \times G$, где G – наибольшее значение яркости, не зависящее от размеров изображения A ; элементы P_{ij} равны количеству пар пикселей a_{cd} и a_{c+ld+m} с яркостями, равными значениям i и j . Элементы P_{ij} вычисляются для всех возможных пар пикселей, разделенных вектором (l, m) , который указывает расположение одного пикселя относительно другого [8]. Для бинарных изображений текстурные признаки на базе МВЯ вычисляются следующим образом (для наглядности введено обозначение $p_{abcd} = p_{ab} + p_{cd}$):

$$f_1 = p_{00}^2 + p_{01}^2 + p_{10}^2 + p_{11}^2; \quad (10)$$

$$f_2 = p_{01} + p_{10} = p_{0110} = 1 - (p_{00} + p_{11}) = 1 - p_{0011}; \quad (11)$$

$$f_3 = \frac{p_{11} - p_{0111}p_{1011}}{\sqrt{p_{0111}p_{0010}p_{1011}p_{0001}}}; \quad (12)$$

$$f_4 = p_{0010}p_{0111} = p_{0010}(1 - p_{0010}); \quad (13)$$

$$f_5 = p_{0011} + 0.5p_{0110} = 0.5(1 + p_{0011}) = p_{0011} + 0.5f_2 = 1 - 0.5f_2; \quad (14)$$

$$f_6 = p_{0110} + 2p_{11} = f_2 + 2p_{11}; \quad (15)$$

$$f_7 = -f_6^2 + f_6 + 2p_{11}; \quad (16)$$

$$f_8 = -\log((p_{00})^{p_{00}}(p_{11})^{p_{11}}(p_{0110})^{p_{0110}}); \quad (17)$$

$$f_9 = -\log((p_{00})^{p_{00}}(p_{01})^{p_{01}}(p_{10})^{p_{10}}(p_{11})^{p_{11}}). \quad (18)$$

1.4 Бинарные признаки

Бинарные признаки могут иметь только два значения, например 0 или 1. При сравнении двух бинарных признаков можно сделать единственный вывод о совпадении или несовпадении значений признаков. Бинарные признаки также могут называться двоичными, булевыми и дихотомическими [9].

1.5 Комбинированные признаки

Комбинированные признаки являются комбинацией геометрических, яркостных и текстурных признаков. Например, в задаче распознавания лица человека используются антропометрические признаки лица человека, специально подобранные для решения задачи распознавания лиц. Антропометрические признаки включают характеристики геометрических, яркостных и текстурных признаков. Например: уголки бровей характеризуются границами текстур волосяного покрова и кожи; области брови и кожи имеют разницу в яркости; брови имеют форму полуовала. Некоторые признаки вычисляются сверткой фильтра в различных точках изображения (x, y) , результатом чего является отклик матрицы изображения в данных точках на форму используемого фильтра. Если в данных точках содержатся признаки объектов, то они отражают информацию о структуре объекта. Например, для вычисления текстурных и яркостных признаков часто применяют набор различных фильтров Габора.

2 Классификация алгоритмов вычисления признаков

2.1 Инвариантность признаков

Условия съемки одних и тех же объектов могут отличаться:

- освещенностью сцены, положением объекта в пространстве;
- ракурсом объектов;
- контрастом изображения, параметрами фотооптики;
- углом съемки объектов;
- масштабом съемки объектов.

Необходимо, чтобы признаки, описывающие объект, были максимально инвариантными к перечисленным параметрам. Признак может быть инвариантным относительно яркостных и геометрических изменений представления объекта на снимке.

Геометрическая инвариантность признаков объекта подразумевает независимость от поворота объекта, изменения масштаба объекта, переноса объекта на изображении, проекции объекта на снимок. Яркостная инвариантность подразумевает независимость от изменений освещенности объекта и изменений контраста изображения.

2.2 Алгоритмы вычисления признаков

Точечные алгоритмы вычисления признаков используют значения отдельных пикселей исходного изображения без учета других. Алгоритм вычисляет признаки, используя значение одного пикселя изображения.

Локальные алгоритмы вычисления признаков используют значения пикселей области изображения. Например, распространенные масочные операции, используемые для вычисления признаков, часто используют области изображений размером от 3×3 до 5×5 пикселей. Например, в свертке используются маски Собеля, Превитта, Робинсона и Лапласиан. Алгоритм SUSAN [11] тоже относится к локальным алгоритмам, поскольку использует окрестность изображения, в которой проверяется область изображения на яркостные изменения, которые обуславливают наличие признаков. Операторы, вычисляющие максимумы и минимумы значений, также относятся к локальным алгоритмам.

К глобальным алгоритмам вычисления признаков (рис. 2) относятся такие алгоритмы, которые используют значения всех пикселей исходного изображения. К алгоритмам данного класса вычисления признаков относятся синусные и косинусные преобразования, преобразование Фурье.

Преобразование Карунена – Лоэва является методом главных компонент (МГК) и выполняет преобразование сигналов в набор некоррелированных коэффициентов [12, 13].

Линейный дискриминантный анализ (ЛДА) [14] используется для снижения размерности и классификации данных.

Анализ независимых компонент (АНК) [15] разработан для разделения смешанных сигналов. АНК дает статистически независимые негауссиановые компоненты путем декорреляции моментов высшего порядка вдобавок к моментам первого и второго порядка статистического распределения. Существует несколько вариантов реализации АНК: FastICA [16], Infomax [17], JADE [18].

Вейвлет-преобразования позволяют вычислять признаки объектов [19]. В работе [20] координаты углов вычисляются с использованием одномерного вейвлет-преобразования функции контура объекта. В работе [21] представлен детектор углов на полутоновых изображениях, основанный на использовании вейвлет-преобразования. Алгоритм вычисляет амплитуду вейвлет-преобразования, которая пропорциональна масштабу расположения углов и некоторых точек края. Алгоритм в работе [22] рассматривает точки изображения, формируемые значениями энергии изображения. Для поиска координат этих точек авторы используют многомасштабную пирамиду. Данный алгоритм определяет много лишних точек на текстурных областях (рис. 3).

Ускорить вычисление признаков можно следующим образом:

- путём ускорения выполнения алгоритмов, с помощью которых выполняется вычисление признаков на изображениях;
- создания фильтров для вычисления определенных типов призна-

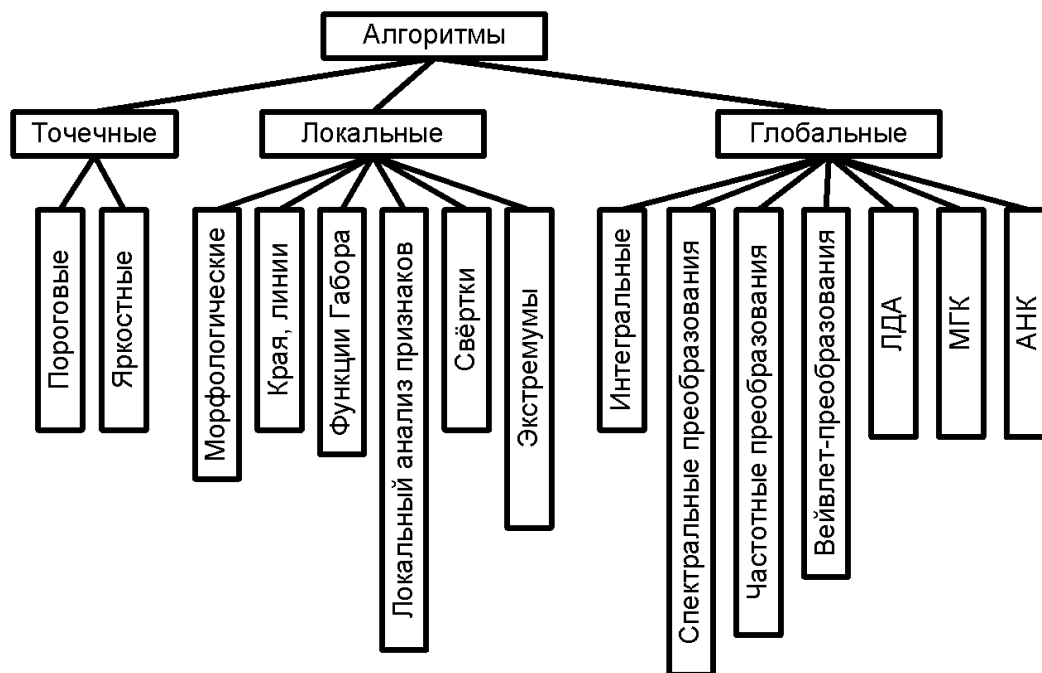


Рис. 2. Алгоритмы вычисления признаков

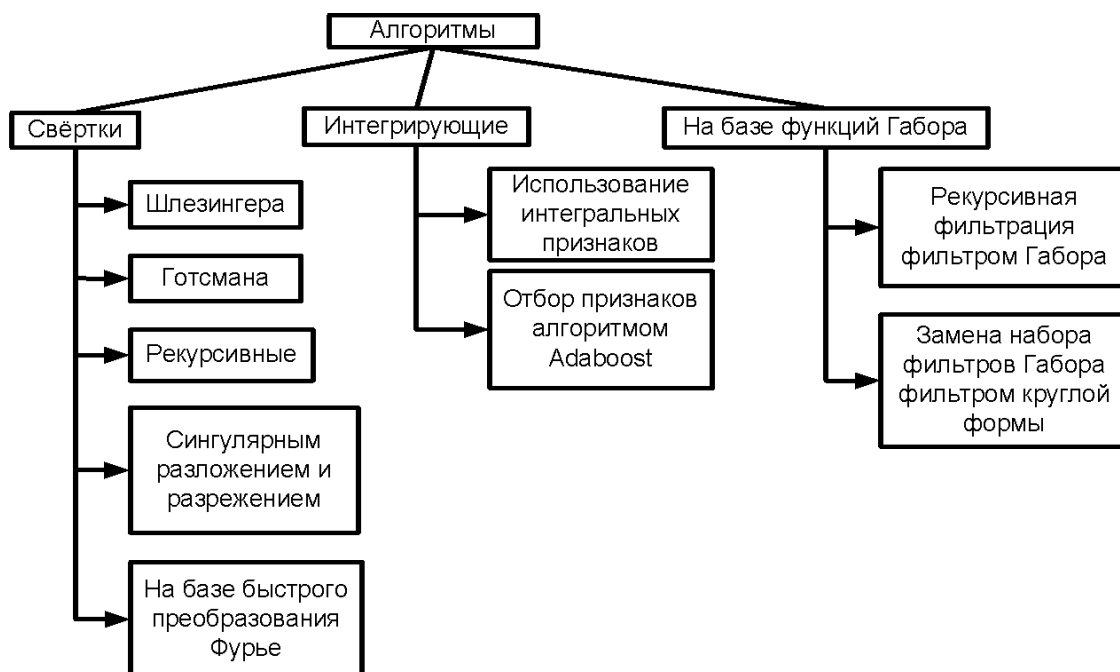


Рис. 3. Основные алгоритмы, используемые в операторах вычисления признаков

ков;

- создания детекторов областей, в которых вычисляются признаки;
- аппроксимации функций, используемых при вычислении признаков;

ков;

- создания специальных рекурсивных алгоритмов вычисления признаков;

- разработки детекторов объектов с помощью обучающего множества изображений, содержащих анализируемые объекты.

3 Алгоритмы предварительного обнаружения объектов

3.1 Обнаружение объектов с использованием интегральных матриц изображений и алгоритма AdaBoost

Пол Виола и Майкл Джонс представили алгоритм устойчивого обнаружения объектов в реальном времени [23]. Данный алгоритм обладает низкой вычислительной сложностью и низким количеством ошибок обнаружения. С помощью последовательности классификаторов в скользящем окне выполняется анализ изображения на наличие обнаруживаемого объекта. Классификаторы из последовательности (от первого до последнего) принимают решения о наличии или отсутствии обнаруживаемого объекта. Если какой-либо классификатор из этой последовательности принимает отрицательное решение, то данная область изображения больше не рассматривается и скользящее окно перемещается к следующей области изображения. Если все классификаторы принимали положительные решения, то данная область может содержать искомый объект.

Интегральными матрицами изображений будем называть такие матрицы, которые вычисляются суммированием элементов исходных изображений по строкам и столбцам.

Классификатор $h_j(d)$ состоит из интегрального признака f_j и порогового значения θ_j :

$$h_j(d) = \begin{cases} 1, & f_j(d) < \theta_j; \\ 0, & \text{иначе} \end{cases} \quad (19)$$

где d – область поиска (скользящее окно) на изображении.

В классификаторах используются разные типы интегральных признаков, четыре типа которых схематически изображены на рис. 4. Интегральный признак *а)* состоит из двух областей (заштрихованной и незаштрихованной), граничащих вертикально. Интегральный признак *б)* также состоит из двух областей, но эти области имеют горизонтальную границу. Значения интегральных признаков *а)* и *б)* вычисляются следующим образом: из суммы значений интегральной матрицы в закрашенной области вычитается сумма значений, находящихся в незакрашенной области. Значение интегрального признака *в)* находится вычитанием сумм пикселей двух

незакрашенных областей из суммы пикселей, находящихся в центральной закрашенной области.

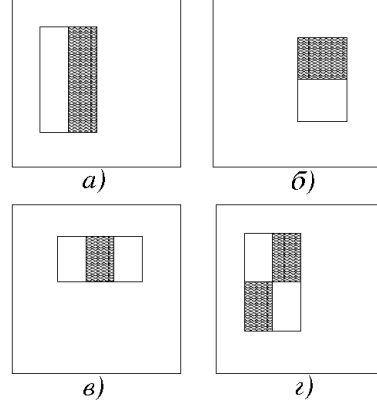


Рис. 4. Четыре типа интегральных признаков

Для быстрого вычисления интегральных признаков исходные изображения преобразуются в интегральные матрицы изображений, на которых и вычисляются интегральные признаки. Использование интегральной матрицы изображения предложено М.И. Шлезингером для ускорения вычисления дискретной свертки [24]. Находим значение пикселя интегральной матрицы изображения с координатами i, j :

$$b(i, j) = \sum_{i' \leq i, j' \leq j} g(i', j'), \quad (20)$$

где g – исходное изображение, b – интегральная матрица изображения.

С помощью рекурсивных зависимостей

$$\begin{aligned} s(i, -1) &= 0; \\ b(-1, j) &= 0; \\ s(i, j) &= s(i, j-1) + g(i, j); \\ b(i, j) &= b(i-1, j) + s(i, j) \end{aligned} \quad (21)$$

интегральная матрица изображения вычисляется путем полного обхода матрицы исходного изображения по строкам и столбцам, где g – исходное изображение, b – интегральная матрица изображения, $s(i, j)$ – сумма строк.

Интегральные матрицы изображений позволяют быстро вычислять суммы значений пикселей на областях любого размера, используя четыре значения интегральной матрицы изображения. Рассмотрим рис. 5. Сумма значений элементов интегральной матрицы изображения области А равна значению элемента матрицы, обозначенного единицей. Сумма элементов области Б равна разности значений элементов интегральной матрицы, обозначенных двойкой и единицей; области В – разности значений элементов интегральной матрицы, обозначенных тройкой и единицей; Г – сумме зна-

чений элементов интегральной матрицы, обозначенных четверкой и единицей, минус значения, обозначенные двойкой и тройкой.

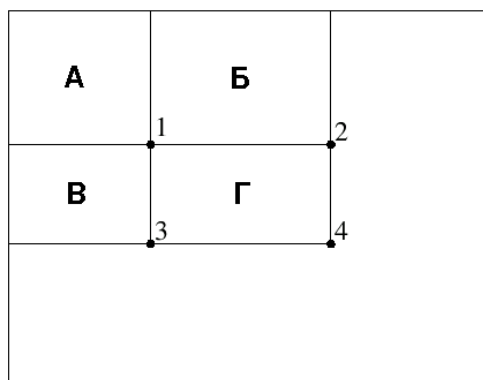


Рис. 5. Схема вычисления значений сумм областей интегральной матрицы изображения

Интегральные признаки чувствительны к наличию краев прямоугольных областей. Для изображения размером 24×24 пикселя максимально возможное количество вычислений вариантов интегральных признаков примерно равно 45 000. Данное количество интегральных признаков является избыточным.

Согласно гипотезе П. Виолы и М. Джонса [23] для формирования последовательности классификаторов необходимо не более 5000 интегральных признаков. Задача отбора лучших интегральных признаков решается с помощью алгоритма Adaboost [25, 26]. Алгоритм Adaboost позволяет улучшить производительность обучающего алгоритма. Имея избыточный набор интегральных признаков и обучающее множество, состоящее из истинных и ложных данных, алгоритм Adaboost позволяет отобрать классификаторы, которые имеют наименьшую ошибку классификации. На каждой итерации алгоритма отбора классификатор h_j принимает решение на обучающем множестве, а алгоритм задает ему вес p_j в зависимости от того, правильно ли было принято решение. После очередной итерации алгоритма отбора значения присвоенных каждому классификатору весов корректируются таким образом, что на последующих итерациях алгоритма отбора рассматриваются те классификаторы, которые неправильно решили задачу классификации на предыдущих итерациях. После выполнения заранее заданного количества итераций алгоритма отбора формируется последовательность, состоящая из лучших классификаторов, т. е. с наименьшей ошибкой классификации. Р. Счэпир, Е. Фреунд и другие доказали, что в алгоритме Adaboost ошибка в классификации результирующего набора классификаторов экспоненциально приближается к нулю при увеличении количества итераций алгоритма отбора классификаторов [25, 26].

Целью алгоритма Adaboost является отбор лучших интегральных признаков, вычисление весов для классификаторов, которые лучше всего разделяют ложные и истинные примеры из обучающего множества. Для каж-

дого классификатора h_j вычисляется такое пороговое значение θ_j , при котором количество неправильно классифицированных примеров из обучающего множества является наименьшим.

Исходным для отбора лучших классификаторов является множество изображений, состоящее из различных изображений обнаруживаемого объекта, т. е. истинных (например, в задаче обнаружения лиц это различные изображения лица человека), и изображений, не содержащих обнаруживаемый объект, т. е. ложных. Представим такое множество вектором $X(i), i = \overline{1...k}$, $X(i)$ – i -е изображение. Вектор $Y(i), i = \overline{1...k}$, содержит соответствующие вектору X значения 0 или 1; $Y(i) = 0$, если $X(i)$ – истинное изображение; $Y(i) = 1$, если $X(i)$ – ложное изображение [23].

Алгоритм отбора лучших интегральных признаков Adaboost

Шаг 1. Установить веса $w_{1,i} = \frac{1}{2m}$, если x_i – истинное изображение; $w_{1,i} = \frac{1}{2n}$, если x_i – ложное изображение; m, n – количество истинных и ложных изображений в обучающем множестве соответственно. Задать параметр T – количество итераций алгоритма. Задать количество отбираемых интегральных признаков.

Шаг 2. На каждой итерации t из $1, \dots, T$ итераций выполнить следующее:

- нормализовать веса

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^k w_{t,j}},$$

где k – размер обучающего множества;

- учитывая, что для каждого интегрального признака j из избыточного набора классификатор h_j принимает решение, используя выражение (19), и на принятие решения влияет один интегральный признак, вычислить значение ошибки классификации:

$$\varepsilon_j = \sum_i^k w_{t,i} |h_j(x_i) - y_i|, x_i \in X;$$

- выбрать классификатор h_t с наименьшей ошибкой ε_t ;
- обновить значения весов:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\varepsilon_i}, \quad \beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}.$$

Итоговый классификатор является взвешенной суммой T классификаторов h_t :

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right),$$

где $\alpha_t = \log \frac{1}{\beta_t}$.

Для устойчивого обнаружения объектов с помощью интегральных признаков необходимо объединить множество классификаторов в последовательность классификаторов. Последовательность состоит из отобранных классификаторов, каждый из которых вычисляет значения интегральных признаков на изображении и выносит решение. На вход каждого классификатора подается область изображения, попадающая в скользящее окно, и на этой области вычисляются значения интегральных признаков. Если значения, вычисленные интегральными признаками, удовлетворяют пороговому ограничению, то область изображения принимается, происходит переход на следующий уровень последовательности, вычисляются признаки на следующем уровне и снова принимается решение об отклонении или принятии области и передаче на последующие уровни. Схематически последовательность, состоящая из n уровней, показана на рис. 6, где И обозначает, что область может содержать обнаруживаемый объект, Л – нет.

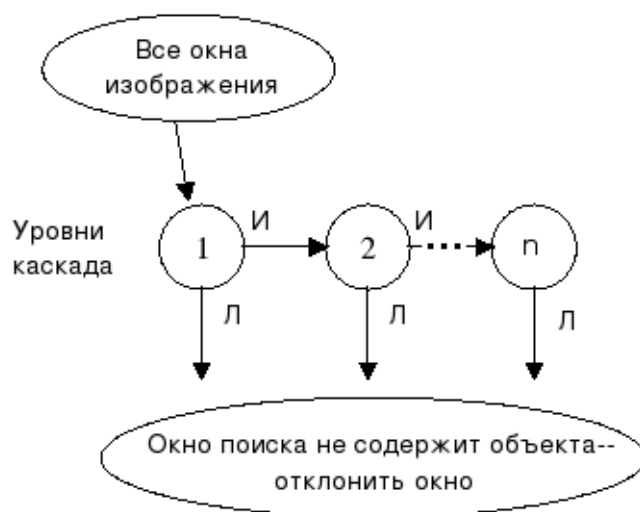


Рис. 6. Схема последовательности классификаторов

Начальные классификаторы последовательности содержат такие интегральные признаки, которые отражают наиболее общие свойства обнаруживаемого объекта. На каждом последующем уровне последовательности производится более точная оценка области изображения. Конечные уровни последовательности содержат интегральные признаки, которые производят самую точную оценку области поиска. Если область изображения была отклонена на начальных уровнях, то она не содержит искомого объекта. Для задачи обнаружения лиц последовательность классификаторов состояла из 32 уровней [23]. Первый уровень содержал два признака; второй – пять; третий, четвертый, пятый – 20; шестой, седьмой – 50; уровни с восьмого по тринадцатый – 100. Все остальные содержали 200 признаков. Большинство ложных областей отбрасывались на начальных уровнях.

Алгоритм обнаружения лица (рис. 7) обладает инвариантностью к масштабу объектов при неизменной скорости их обнаружения. Для об-

наружения объектов различного масштаба необходимо масштабирование интегральных признаков. Поскольку интегральные признаки вычисляются из значений сумм областей изображения и для вычисления одной суммы требуется не более четырех значений элементов интегральной матрицы изображения, то изменение масштаба интегральных признаков не увеличит времени их вычисления на интегральной матрице изображения.



Рис. 7. Результат применения алгоритма обнаружения лица

Недостатком алгоритма является большое время отбора интегральных признаков, которое линейно увеличивается при увеличении размера обучающего множества [23], но отбор классификаторов необходимо провести только один раз. При размере обучающего множества в 14 916 изображений лиц размером 24×24 отбор необходимого количества классификаторов на компьютере AlphaStation XP900 466 МГц выполнялся несколько недель.

Данный алгоритм обладает высокой скоростью обнаружения объектов и устойчивостью. На компьютере с процессором Pentium 3 700 МГц алгоритм обнаруживает объекты на изображении размером 384×288 за 0,04 с. Чем больше размер обучающего множества, тем достовернее будет обнаружение объектов, выше точность и ниже ошибка обнаружения.

Для создания качественных классификаторов рекомендуемый размер обучающего множества изображений объекта должен составлять не менее 3000 изображений. Количество ложных изображений, не содержащих объект, должно быть в 1,5 раза больше.

Поскольку время отбора классификаторов линейно зависит от размера обучающего множества и для создания качественных классификаторов необходим относительно большой размер обучающего множества, то возникает задача снижения времени отбора классификаторов. Данную задачу можно решать двумя способами:

- уменьшить количество классификаторов и увеличить обучающее множество;
- уменьшить обучающее множество и увеличить количество классификаторов.

Уменьшение обучающего множества негативно сказывается на качестве классификаторов при их неизменном количестве, но при уменьшении обучающего множества можно увеличить количество классификаторов.

Поскольку отбор классификаторов в данном алгоритме производится с использованием заданного обучающего множества изображений объекта, то возможно обнаружение произвольных объектов на изображениях путем выбора соответствующего обучающего множества. Также необходимо создать обучающее множество изображений оптимального размера для отбора достаточного количества классификаторов и обеспечения их качества.

3.2 Использование детектора SUSAN для вычисления признаков объектов

Часто для вычисления краев объектов на изображениях применяются фильтры Собеля, Робертса, Превитта и др. [12]. Чтобы вычислить края объекта на изображении, выполняется свертка масок этих фильтров с изображениями. Обычно эти маски имеют размер 3×3 .

Детектор SUSAN [11] используется для вычисления краев и углов объектов на изображениях. Детектор использует скользящее окно круглого вида. Окно состоит из центрального элемента \vec{r}_0 (ядра детектора) и остальной части \vec{r} .

Окно последовательно накладывается на все точки изображения, и для каждой такой точки выполняется сравнение уровней яркостей пикселей окна детектора SUSAN $I(\vec{r})$ с яркостью центрального элемента окна $I(\vec{r}_0)$ – ядра детектора. Сравнение выполняется при помощи зависимостей

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1, & |I(\vec{r}) - I(\vec{r}_0)| \leq t; \\ 0, & |I(\vec{r}) - I(\vec{r}_0)| > t, \end{cases} \quad (22)$$

где \vec{r}_0 – расположение ядра детектора на изображении; \vec{r} – расположение любой другой точки внутри окна детектора на изображении; $I(\vec{r}_0)$ – значение яркости пикселя \vec{r}_0 ; t – порог разности яркостей. График функции, заданной выражением (22), изображен на рис. 8 и помечен буквой а.

Результат сравнения пикселей, находящихся внутри окна детектора, вычисляется следующим образом:

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0). \quad (23)$$

Порог t задает минимальный контраст признаков, которые будут определены, и максимальный уровень шума, который будет проигнорирован. Затем n сравнивается с фиксированным порогом g («геометрическим порогом» [11]), R присваивается значение $3/4n_{max}$, где n_{max} является максимальным значением, которое может принимать n на изображении. Выра-

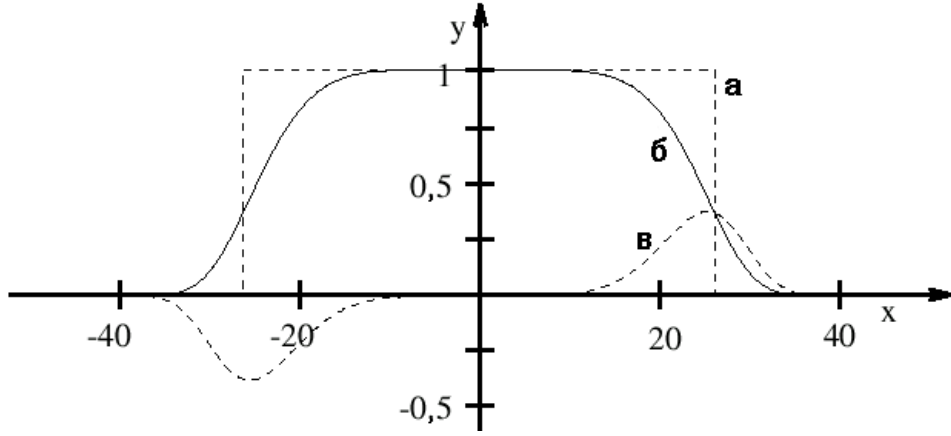


Рис. 8. График функции подобия $c(\vec{r}, \vec{r}_0)$

жение для вычисления краев записывается следующим образом:

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0), & n(\vec{r}_0) < g, \\ 0, & n(\vec{r}_0) \geq g, \end{cases} \quad (24)$$

где R – результат вычисления краев.

Вычисление величины c в формуле (22) дает хорошие результаты, но более устойчивые результаты дает зависимость

$$c(\vec{r}, \vec{r}_0) = e^{-\left(\frac{I(\vec{r}) - I(\vec{r}_0)}{t}\right)^6}. \quad (25)$$

График функции (25) изображен на рис. 8 и помечен буквой б.

Рассмотрим одномерный сигнал, значение которого монотонно возрастает. Можно проиллюстрировать, что минимальное значение длины ядра (в двумерном случае – площади ядра) будет эквивалентно геометрическому расположению края. На рис. 9 изображены одномерная функция и три расположения одномерных масок SUSAN.

На рис. 10 и 11 показаны исходные изображения и результаты вычисления краев детектором SUSAN, главными достоинствами которого являются сравнительно высокие скорость вычисления и точность.

Поскольку детектор SUSAN не использует свертку, то скорость его работы выше, чем скорости работы обычных детекторов [11]. Качество вычисления признаков на зашумленных изображениях падало значительно меньше, чем у иных детекторов.

При использовании детектора SUSAN для вычисления краев и углов объектов на изображении значение порога t находится в диапазоне от 10 до 50. При вычислении краев объектов рекомендуемым значением порога является $t = 20$. При вычислении углов объектов рекомендуемым значением порога является $t = 10$.

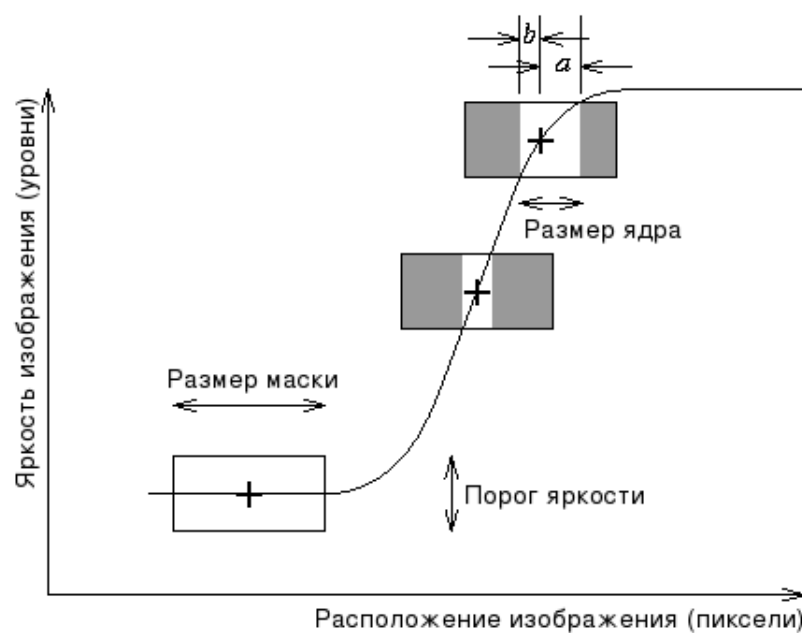


Рис. 9. Результат применения одномерной маски детектора SUSAN на одномерном сигнале

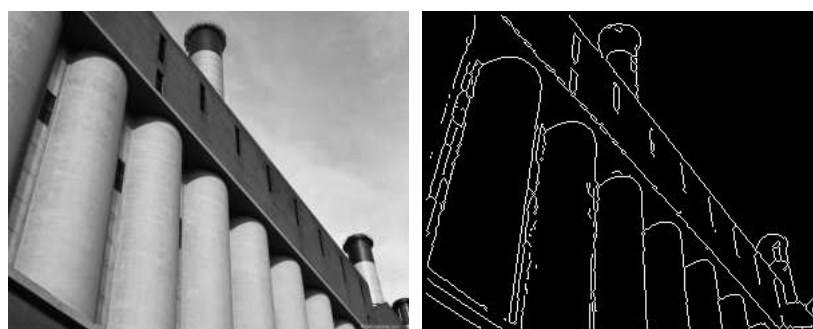


Рис. 10. Исходное изображение и результат вычисления краев детектором SUSAN (значение порога детектора SUSAN $t=20$; s вычисляется по формуле (25))

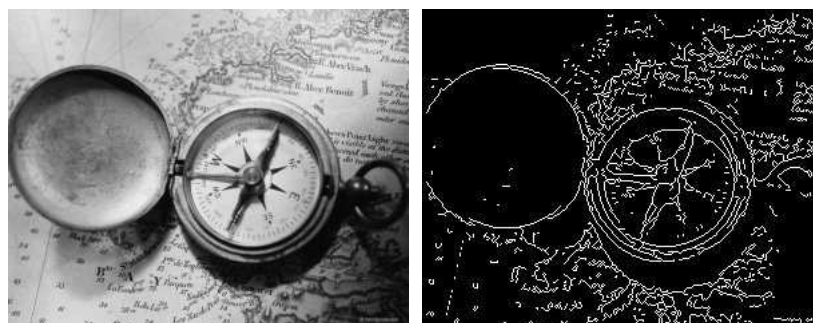


Рис. 11. Исходное изображение компаса и результат вычисления краев детектором SUSAN (значение порога детектора SUSAN $t=40$; s вычисляется по формуле (25))

4 Алгоритмы ускорения вычисления дискретной свертки

Свертка является одной из базовых операций обработки изображений. Это локальная операция, которая часто используется для вычисления признаков объектов на изображениях. Сложность свертки составляет $O(MNp^2)$, где M, N – размеры изображения, $p \times p$ – размер ядра фильтра. Для некоторых фильтров задачи оптимизации дискретной свертки решены [27, 28], но зачастую эти решения существуют только для фильтра с фиксированными параметрами, что существенно ограничивает область применения таких алгоритмов.

Необходимо отметить, что свертка на основе быстрого преобразования Фурье быстрее обычной свертки только при использовании ядер фильтров с большими размерами (с примерным размером больше 10×10). Ядра фильтров, имеющие размеры более 10×10 , реже используются при обработке изображений.

При обработке изображений ядрами фильтров размером меньше чем 10×10 более быстрым алгоритмом является дискретная свертка, нежели свертка, использующая БПФ. Поскольку в обработке изображений преимущественно используются фильтры таких размеров, то возникает необходимость в разработке новых алгоритмов ускорения дискретной свертки.

Основой интегрально-дифференциального алгоритма вычисления свертки [24] является преобразование дискретного изображения интегрированием, а ядра фильтра – дифференцированием. Интегрирование и дифференцирование производятся по строкам и столбцам. Поскольку свертка инвариантна к операциям интегрирования и дифференцирования, то после аппроксимирования и дифференцирования ядро фильтра упрощается и становится разреженным. При свертке с разреженным ядром фильтра не используются элементы, содержащие нулевые значения, и количество операций сложения и умножения снижается.

Алгоритм сингулярного разложения может быть использован для разделения функций двух переменных. Для ускорения свертки Готсман предложил сингулярное разложение ядра фильтра на векторы [29]. При заранее известном аналитическом задании фильтра необходимо разложить ядро фильтра на вектор-столбец и вектор-строку. Не все фильтры можно разложить на векторы [29]. После данного разложения выполняется свертка изображения последовательно с этими двумя векторами.

В алгоритме сингулярного разложения и разрежения (СРР) векторов производится сингулярное разложение ядра фильтра на вектор-столбец и вектор-строку. Затем выполняется аппроксимация этих векторов интервалами M длиной k , где первому элементу каждого интервала M присваивается среднее значение интервала, а остальные значения обнуляются и при свертке не используются. После данных преобразований выполняется последовательная свертка изображения с этими двумя векторами, используя лишь начальные ненулевые элементы в каждом интервале. Таким образом,

при свертке сокращается количество используемых элементов векторов и количество операций умножения и сложения.

4.1 Интегрально-дифференциальный алгоритм вычисления свертки

Дискретная свертка функции f с фильтром g в точке (n_1, n_2) записывается следующим образом:

$$(f \otimes g)(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)g(n_1 - k_1, n_2 - k_2), \quad (26)$$

где f – функция; g – фильтр.

М.И. Шлезингер [24] доказал, что операцию свертки можно преобразовать:

$$f \otimes g = f^{(n)} \otimes g^{(-n)}, \quad (27)$$

где f – функция яркости (изображение); g – ядро фильтра; n – положительное целое число; (n) – порядок интегрирования; $(-n)$ – порядок дифференцирования; \otimes – операция свертки.

Вычисляя производную аппроксимированного ядра фильтра, данное ядро можно упростить до такой степени, что большинство элементов станут равны нулю. Проинтегрированное изображение $f^{(n)}$ будем называть интегральной матрицей изображения. При свертке интегральной матрицы изображения с преобразованным ядром фильтра количество операций на один элемент матрицы меньше, чем при свертке с обычным ядром фильтра, так как отпадает необходимость использовать нулевые элементы ядра фильтра.

Для снижения вычислительных затрат алгоритмы интегрирования и дифференцирования реализованы рекурсивно. Изображение необходимо преобразовать в интегральную матрицу изображения в соответствии с рекурсивными выражениями

$$\begin{cases} s(i, j) = s(i, j - 1) + f(i, j); \\ z(i, j) = y(i - 1, j) + s(i, j), \end{cases} \quad (28)$$

где f – изображение, z – интегральная матрица изображения, s – рабочая матрица.

Продифференцированное ядро фильтра для ускоренной свертки вычисляется в соответствии с рекурсивными выражениями

$$\begin{cases} s(i, j) = h(i, j) - h(i, j - 1); \\ r(i, j) = s(i, j) - s(i - 1, j), \end{cases} \quad (29)$$

где h – ядро фильтра; r – результирующее ядро фильтра для свертки с интегральной матрицей изображения; s – рабочая матрица.

Настоящий алгоритм применим к любым фильтрам, но перед дифференцированием необходимо выполнить аппроксимацию ядра фильтра.

Интегрально-дифференциальный алгоритм вычисления свертки

Шаг 1. Вычислить интегральную матрицу изображения в соответствии с выражениями (28).

Шаг 2. Выполнить аппроксимацию ядра фильтра прямоугольными либо квадратными областями заранее заданной размерности. Размеры области аппроксимации должны быть меньше размеров ядра фильтра. Данная операция необходима для того, чтобы после дифференцирования ядро фильтра упростилось.

Шаг 3. Выполнить дифференцирование аппроксимированного ядра фильтра с помощью выражений (29). Элементы ядра фильтра становятся нулевыми из-за того, что после аппроксимации матрицы областями получается мало перепадов яркостей пикселей матрицы (перепады имеются только на границах областей аппроксимации изображения).

Шаг 4. Выполнить свертку интегральной матрицы изображения с продифференцированным ядром фильтра, не учитывая нулевые элементы продифференцированного ядра фильтра.

После дифференцирования ядро фильтра становится разреженным, причем степень разреженности зависит от формы фильтра (рис. 12).

На рис. 13 представлен пример ядра фильтра (размер ядра фильтра – 33×33) и его аппроксимированный вариант блоками 5×5 пикселей, а на рис. 14 – интегральной матрицы изображения.

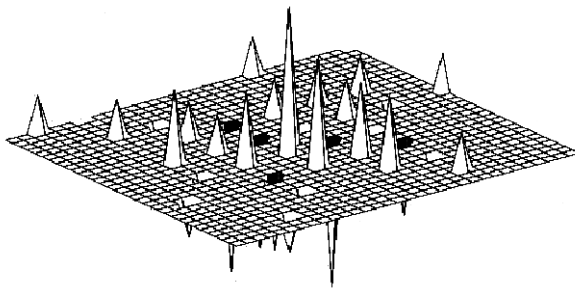


Рис. 12. Пример продифференцированного ядра фильтра

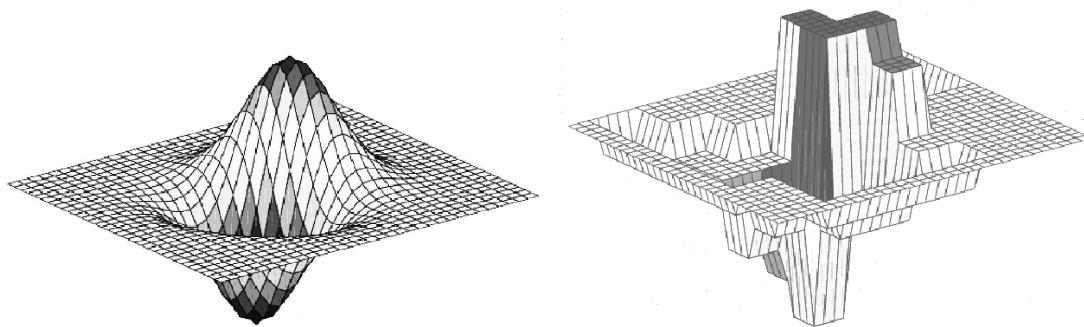


Рис. 13. Пример ядра фильтра и его аппроксимированного варианта

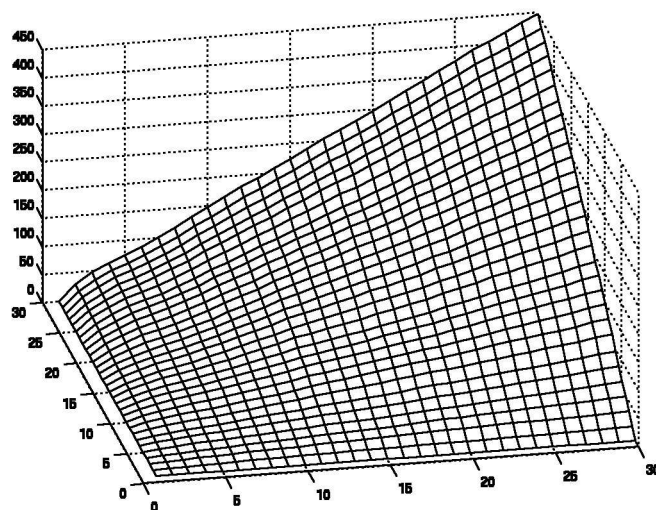


Рис. 14. Пример интегральной матрицы изображения

Критическим моментом в данном алгоритме является выбор размеров области аппроксимации. Чем больше размер области аппроксимации, тем меньше границ перепадов в уровнях яркостей между областями аппроксимации и ненулевых элементов в матрице.

Алгоритм апробирован в среде Matlab 6.5. На примере полутонного изображения saturn.tif размером 256×256 пикселей и при размерах области аппроксимации 2×2 элементов ядра фильтра Габора размером 8×8 были получены следующие результаты дискретных сверток, выраженные в секундах. При свертке с использованием быстрого преобразования Фурье (БПФ) время составило 0,83 с, а при ускоренной (с учетом интегрирования, дифференцирования и аппроксимации) – 0,33 с на компьютере с процессором Pentium 3 1400 МГц. Корреляция результирующих изображений равна 0,9958. Оптимальные размеры области аппроксимации фильтров находятся в диапазоне от 2×2 до 10×10 . Рекомендуемой областью аппроксимации является область размером 5×5 .

Реализация алгоритма позволяет получить выигрыш по времени приблизительно на 30%. Для получения лучших результатов необходимо снизить временные затраты на поиск ненулевых элементов ядра фильтра и на выбор значений пикселей из ядра фильтра и свертываемого изображения.

Поскольку фильтры простой формы, например гауссиан, содержат мало экстремумов и точек перегиба, то преобразованное ядро фильтра получается сильно разреженным. При использовании алгоритма снижается время вычисления операций сложения и умножения из-за относительно небольшого количества ненулевых элементов продифференцированного ядра фильтра. Для фильтров сложного вида возрастает количество ненулевых значений матрицы фильтра, а соответственно и время на поиск ненулевых элементов фильтра и выбор значений из матриц изображения и фильтра.

При увеличении размеров области аппроксимации, применяемой к ядру

фильтра, скорость свертки увеличивается, но падает качество фильтрации.

4.2 Сингулярное разложение фильтра для ускорения свертки

Для ускорения свертки К. Готсман предложил сингулярное разложение ядра фильтра на вектор-строку и вектор-столбец [29]. После разложения ядра необходимо последовательно выполнить свертку с этими двумя векторами. Такая свертка требует меньше времени, нежели свертка с целым ядром фильтра.

Если известно аналитическое задание фильтра, то возможно разложение ядра фильтра на векторы. Обычно малое количество функций являются разделимыми. Функция двух переменных $M(x, y)$ является разделимой, если ее можно преобразовать следующим образом: $M(x, y) = u(x)^T v(y)$.

Свертка обладает свойством

$$f_1 \otimes f_2 = f_1 \otimes (uv) = (f_1 \otimes u) \otimes v, \quad (30)$$

где f_1, f_2 – некоторые функции, $f_2 = uv$.

Если разложение ядра фильтра на векторы является возможным, то ядро фильтра M раскладывается следующим образом:

$$M = \sum_{i=1}^m u_i^T \sigma v_i \quad (31)$$

и в матричном виде

$$M = U^T S V, \quad (32)$$

где M – матрица размером $n \times n$; u^T, v – действительные векторы-столбцы; σ – матрица положительных скаляров; U^T, S и V – действительные матрицы размером $n \times n$; S – диагональная, U, V – ортонормальные.

Алгоритм ускорения свертки с использованием сингулярного разложения фильтра

Шаг 1. Выполнить разложение матрицы на вектор-строку и вектор-столбец [30].

Шаг 2. Последовательно выполнить свертку с этими векторами.

Например, простые двумерные функции Гаусса

$$g(x, y) = e^{-\frac{(x^2+y^2)}{\sigma_x \sigma_y}} \quad (33)$$

раскладываются на два вектора по переменным x и y :

$$e^{-(x^2+y^2)} = e^{-\frac{x^2}{\sigma_x}} e^{-\frac{y^2}{\sigma_y}}. \quad (34)$$

Например, двумерный гауссиан G (рис. 15) раскладывается на векторы g_x и g_y^T , изображенные на рис. 16.

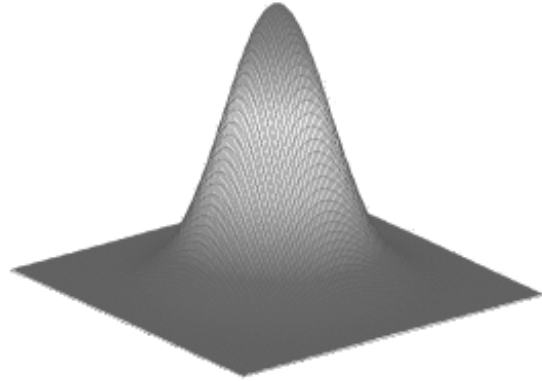


Рис. 15. Изображение гауссиана G

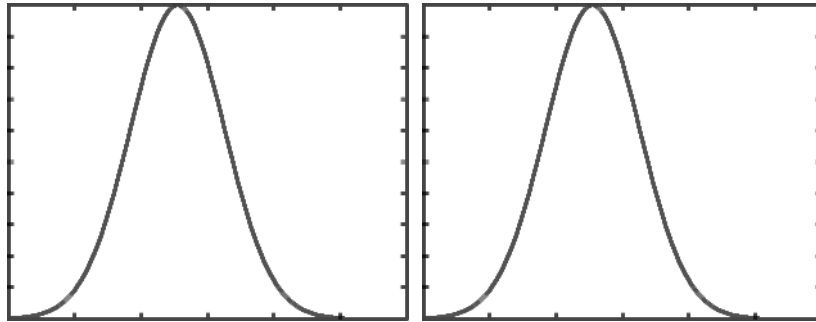


Рис. 16. Графические отображения векторов g_x и g_y

При этом верно равенство

$$G = g_x g_y^T, \quad (35)$$

где $g_x = e^{-\frac{x^2}{\sigma_x}}$; $g_y = e^{-\frac{y^2}{\sigma_y}}$.

Рассмотрим другой пример. Пусть матрица M имеет вид

$$M = \begin{pmatrix} 15 & 45 & 30 & 5 \\ 21 & 63 & 42 & 7 \\ 9 & 27 & 18 & 3 \\ 27 & 81 & 54 & 9 \end{pmatrix}.$$

Она раскладывается на два вектора: $M_1 = (5, 7, 3, 9)$ и $M_2 = (3, 9, 6, 1)$. Свертка с этими двумя векторами требует меньшего количества операций умножения и сложения, чем при использовании целой матрицы M .

4.3 Алгоритм сингулярного разложения и разрежения векторов

В процессе исследований был разработан новый алгоритм ускорения вычисления дискретной свертки. Алгоритм заключается в сингулярном разложении ядра фильтра на векторы u и v , аппроксимации этих векторов, их разрежении и последовательной свертке изображения с этими векторами. Поскольку эти разреженные векторы содержат нулевые значения, которые не учитываются при свертке, то свертка с этими векторами требует меньшего количества операций умножения и сложения.

С помощью сингулярного разложения матрицы по формуле

$$M = \sum_{i=1}^n u_i^T \sigma v_i \quad (36)$$

вычисляются векторы u и v , которые аппроксимируются с заранее выбранным интервалом H длины k . Результаты аппроксимации сохраняются в векторы u' и v' . В каждом интервале $H_j = \{h_j, h_{j+1}, \dots, h_{j+k}\}$ векторов u' и v' начальному элементу H_j устанавливается интервальное значение, полученное при аппроксимации, а остальным элементам интервала $h_{j+1}, h_{j+2}, \dots, h_{j+k}$ – нулевые значения, которые при свертке не учитываются.

Далее выполняется свертка с векторами u' и v' без учета нулевых элементов:

$$I \otimes M \approx (I \otimes u') \otimes v', \quad (37)$$

где I – изображение.

Основу предлагаемого алгоритма составляют аппроксимация векторов разложения ядра фильтра u и v , их разрежение и последовательная свертка изображения с этими векторами.

Алгоритм CRR

Шаг 1. Разложить ядро фильтра на векторы u и v с помощью сингулярного разложения [30].

Шаг 2. Аппроксимировать векторы u и v с заранее выбранным интервалом H длины k . Сохранить результаты аппроксимации в векторах u' и v' .

Шаг 3. В каждом интервале H векторов u' и v' начальному элементу H_j установить среднее интервальное значение, полученное при аппроксимации, а остальным элементам интервала $h_{j+1}, h_{j+2}, \dots, h_{j+k}$ – нулевые значения.

Шаг 4. Выполнить последовательную свертку векторов u' и v' с изображением, не учитывая нулевые элементы в векторах.

Новым в данном алгоритме является то, что кроме разложения ядра фильтра на векторы свертка ускоряется за счет аппроксимации, разрежения векторов разложения и использования не всех элементов разреженных

векторов. Степень разреженности задается значением длины k интервала аппроксимации H . Качество свертки зависит от степени разреженности векторов. Чем меньше интервал аппроксимации, тем выше качество свертки и больше времени требуется на её выполнение.

4.4 Экспериментальный анализ алгоритмов свертки

В таблице представлено время выполнения сверток тремя алгоритмами: интегрально-дифференциальным, сингулярного разложения фильтра для ускорения свертки и СРР. В качестве данных для сравнения затраченного времени при фильтрации выбраны изображения размером 200×200 , 400×400 , 600×600 ; размеры ядра фильтра 50×50 , 100×100 ; шаг аппроксимации 5 и 10. Изображения задавались с помощью генератора случайных чисел. В качестве фильтров использовались фильтры Гаусса. Фильтрация производилась на компьютере с процессором Pentium 4 1200 МГц.

Таблица

Результаты сравнения скоростей алгоритмов

Размер изображения, $N \times N$	Размер ядра, $M \times M$	Шаг аппроксимации	Обыкновенная свертка, с	Интегрально-дифференциальный алгоритм, с	Алгоритм сингулярного разложения, с	СРР, с	Корреляция обычной свертки и СРР
200	50	5	0,55	0,49	0,06	0,05	0,9930
200	50	10	0,55	0,49	0,06	0,06	0,9752
200	100	5	2,86	2,14	0,11	0,11	0,9981
200	100	10	2,86	2,47	0,11	0,06	0,9793
400	50	5	2,25	2,14	0,38	0,27	0,9793
400	50	10	2,25	2,19	0,16	0,16	0,9726
400	100	5	11,80	9,39	0,66	0,61	0,9961
400	100	10	11,80	10,06	0,66	0,55	0,9410
600	50	5	5,38	5,05	0,44	0,38	0,9987
600	50	10	5,38	4,89	0,50	0,44	0,9913
600	100	5	26,03	20,43	1,49	1,43	0,9983
600	100	10	26,03	20,44	1,42	1,37	0,9958

Интегрально-дифференциальный алгоритм позволяет получать разреженные ядра фильтров, содержащие ненулевые элементы, что снижает вычислительную нагрузку ЭВМ, используя при свертке только ненулевые элементы. При применении алгоритма максимальная скорость получается, если заранее известно расположение ненулевых элементов и не тратятся время и память на хранение координат ненулевых элементов и их поиск.

Алгоритм сингулярного разложения ядра фильтра для ускорения свертки дает увеличение скорости минимум на один порядок в зависимости от размера матрицы изображения и ядра фильтра. Не все матрицы могут

быть разложены на векторы, что дает ограничение на использование этого алгоритма. Например, в алгоритме Шлезингера возможно использование любых фильтров.

СРР быстрее алгоритма сингулярного разложения фильтра для ускорения свертки, что обусловлено использованием лишь некоторых ненулевых элементов из аппроксимированных векторов разложения. Но данный алгоритм, в отличие от прямого использования векторов, аппроксимирует эти векторы, и в связи с этим происходит потеря качества свертки. При увеличении шага аппроксимации увеличивается скорость и снижается качество результата свертки.

Легко выполнить разложение фильтров (например, гауссиана, фильтра Габора), зная их аналитическое представление. Труднее сделать разложение матриц фильтров, не зная их аналитического задания.

При больших размерах матрицы фильтра (от 10×10) для повышения скорости фильтрации необходимо применение быстрого преобразования Фурье.

Рекомендуемым шагом аппроксимации в интегрально-дифференциальном и в предлагаемом алгоритмах является величина, равная 3.

5 Фильтры Габора для вычисления яркостных и текстурных признаков

5.1 Использование фильтров Габора для анализа цифровых изображений

Функции Габора были введены в 1946 г. для анализа одномерных функций [10]. Даугман развил данные функции и применил для анализа текстур на изображениях [31].

Марселиа [32] и Даугман [33, 34] дали теоретическое обоснование того, что простые ячейки в зрительной коре мозга млекопитающих могут быть смоделированы с помощью элементарных функций Габора. После открытия зрительной системы в коре головного мозга млекопитающих количество исследований и экспериментов с функциями Габора резко возросло. Двумерные функции Габора были обобщены и преобразованы Даугманом в локальные пространственно-частотные фильтры.

Оригинальные одномерные фильтры Габора вычисляются с помощью гауссиана и модулирующей волновой функции.

Ядро двумерного фильтра Габора задается следующим образом:

$$g(x, y, \omega, \theta) = \frac{\omega^2}{\pi \sigma_x \sigma_y} e^{-(\frac{\omega^2}{\sigma_x^2} x'^2 + \frac{\omega^2}{\sigma_y^2} y'^2)} e^{j2\pi \omega x'}, \quad (38)$$

где $x' = x \cos \theta + y \sin \theta$; $y' = -x \sin \theta + y \cos \theta$; ω – период синусоидальной составляющей фильтра; θ – поворот фильтра; σ_x – размер ядра фильтра по оси x ; σ_y – размер ядра фильтра по оси y ; $j = \sqrt{-1}$. Фильтр состоит из

действительной и мнимой частей. Действительную часть фильтра образует косинусоидальная составляющая, а мнимую – синусоидальная (рис.17). Действительная часть имеет похожий вид.

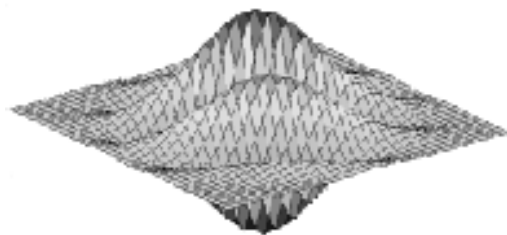


Рис. 17. Пример мнимой части ядра фильтра Габора со следующими параметрами:
 $\sigma_x = \sigma_y = 7; f = 3\pi; \theta = \frac{\pi}{3}$

После свертки изображения с фильтром на границах текстур образуются локальные экстремумы, или резкие перепады значений. Причем максимальный перепад будет на тех границах текстур, пространственная ориентация которых наиболее близка к пространственной ориентации фильтра.

Если изображение не имеет резких перепадов яркостей, то после обработки изображения фильтром Габора значения уровней яркостей всего изображения увеличатся, уменьшатся или останутся прежними в зависимости от параметров фильтра.

Основной недостаток использования данных фильтров – это вычислительная сложность, поскольку в каждой точке необходимо выполнять свертку с ядром фильтра.

5.2 Минимизация набора фильтров Габора

Для того чтобы наиболее полным образом провести анализ изображения, используется набор фильтров с различными параметрами. Например, набор включает фильтры с различными углами пространственной ориентации, различными размерами и различными значениями параметров σ_x, σ_y . Часто при анализе изображений используются наборы ядер фильтров, имеющие пять различных размеров σ и восемь различных пространственных ориентаций θ [35, 36] – всего 40 ядер фильтров (рис. 18).

После свертки со всеми фильтрами выполняется анализ полученных результатов. Поскольку дискретная свертка требует времени, то возрастает суммарно затраченное время на фильтрацию набором ядер фильтров.

Можно использовать фильтры Габора круглого вида. Тип данного фильтра позволяет заменить фильтры с различной пространственной ориентацией одним фильтром круглого вида.

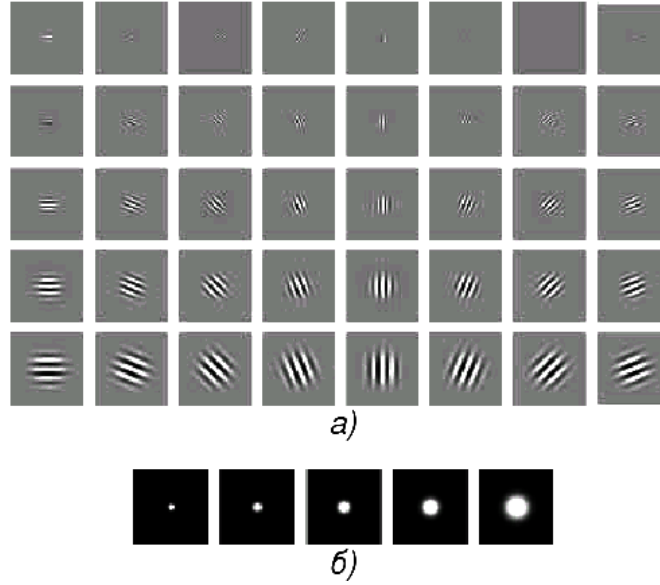


Рис. 18. Набор ядер фильтров Габора с изменяющимися значениями размера σ и пространственной ориентации θ :
 а) действительные части ядер фильтров Габора пяти разных размеров и восьми разных ориентаций; б) амплитуды ядер фильтров Габора пяти разных размеров

Ядро фильтра Габора круглого вида (рис. 19) задается выражением

$$cg(x, y, F) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} e^{2\pi jF(\sqrt{x^2+y^2})}, \quad (39)$$

где F – частота синусоидальной составляющей фильтра круглого вида.

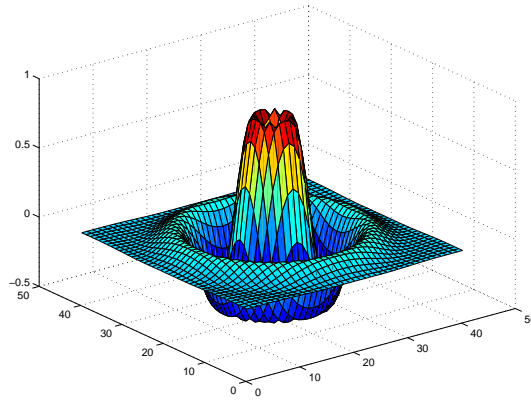


Рис. 19. Ядро фильтра Габора круглого вида (40×40 ,
 $\sigma = 7$, $F=0,2$)

Преимущество данного фильтра круглого вида состоит в том, что не нужно использовать различные пространственные ориентации фильтра. В этом случае набор фильтров Габора круглого вида состоит из пяти фильтров различных размеров.

6 Ускорение вычисления свертки через рекурсию

Для снижения времени вычисления признаков и анализа изображений создаются рекурсивные алгоритмы. Рекурсивные алгоритмы дают приемлемую ошибку реализации алгоритма. Далее рассмотрены рекурсивные алгоритмы свертки с гауссианом [27] и функцией Габора [28].

Данные рекурсивные алгоритмы позволяют выполнять свертку, требующую $O(N)$ операций произведения и сложения для сигнала, состоящего из N отсчетов. Сложность этих алгоритмов не зависит от таких параметров, как σ для гауссиана и σ и ω (период синусоидальной составляющей) для функции Габора. Рекурсивный алгоритм свертки с функцией Габора, кроме прямой фильтрации, позволяет выполнять обратную фильтрацию [28]. Данные алгоритмы являются самыми быстрыми из известных алгоритмов свертки с гауссианом и функцией Габора. Поскольку вычислительная сложность алгоритмов составляет $O(N)$, то данные алгоритмы выполняют свертку быстрее, чем свёртка с использованием БПФ.

6.1 Рекурсивный алгоритм свертки с гауссианом

В работе [27] представлен рекурсивный алгоритм выполнения свертки с гауссианом взамен обыкновенной свертки, или свертки с использованием БПФ. Сложность данного алгоритма свертки составляет 12 операций произведения и сложения на один элемент изображения, что меньше, чем выполнение свертки с использованием БПФ и, тем более, обыкновенной свертки. Алгоритм состоит из двух частей – прямой и обратной рекурсий. Входными данными для прямой рекурсии является массив данных, обозначенный переменной *in*; входными данными для обратной рекурсии – массив данных *w*; результатом алгоритма – массив данных *out*.

Прямая рекурсия задается выражением

$$v(n) = B * in(n) + (b_1 * v(n-1) + b_2 * v(n-2) + b_3 * v(n-3))/b_0. \quad (40)$$

Обратная рекурсия задается выражением

$$out(n) = B * v(n) + (b_1 * out(n+1) + b_2 * out(n+2) + b_3 * out(n+3))/b_0, \quad (41)$$

где n – индекс рекурсивно вычисляемого элемента в массиве.

Значения b_0, b_1, b_2, b_3, B вычисляются следующими выражениями:

$$\begin{aligned} b_0 &= 1,57825 + 2,44413 * q + 1,4281 * q^2 + 0,422205 * q^3; \\ b_1 &= 2,44413 * q + 2,85619 * q^2 + 1,26661 * q^3; \\ b_2 &= -(1,4281 * q^2 + 1,26661 * q^3); \\ b_3 &= 0,422205 * q^3; \\ B &= 1 - (b_1 + b_2 + b_3)/b_0. \end{aligned} \quad (42)$$

Значение q вычисляется выражением

$$q = \begin{cases} 0,98711 * \sigma_0 - 0,96330, & \sigma_0 \geq 2,5; \\ 3,97156 - 4,14554 * \sqrt{1 - 0,26891 * \sigma_0}, & 0,5 \leq \sigma_0 \leq 2,5, \end{cases} \quad (43)$$

где σ_0 – числовое значение параметра σ фильтра Гаусса.

Данные выражения прямой и обратной рекурсии обобщаются к двумерному случаю для возможности обработки изображений.

Алгоритм рекурсивной свертки с гауссианом

Шаг 1. Вычислить значения q, b_0, b_1, b_2, b_3, B , используя выражения (42), (43).

Шаг 2. Выполнить проход матрицы изображения прямой и обратной рекурсиями, используя выражения (40), (41).

6.2 Рекурсивный алгоритм свертки с функцией Габора

В работе [28] представлен алгоритм рекурсивной свертки с функцией Габора.

Прямая рекурсия задается выражением

$$v(n) = in(n) - ((b_1 * e^{i\Omega_0} * v(n-1)) + (b_2 * e^{i2\Omega_0} * v(n-2)) + (b_3 * e^{i3\Omega_0} * v(n-3))). \quad (44)$$

Обратная рекурсия задается выражением

$$out(n) = B * v(n) - ((b_1 * e^{-i\Omega_0} * out(n+1)) + (b_2 * e^{-i2\Omega_0} * out(n+2)) + (b_3 * e^{-i3\Omega_0} * out(n+3))), \quad (45)$$

где n – индекс рекурсивно вычисляемого элемента в массиве, Ω_0 – значение периода синусоидальной составляющей ω функции Габора.

Значения $scale, b_0, b_1, b_2, b_3, B$ вычисляются следующими выражениями:

$$\begin{aligned} scale &= (m_0 + q) * (m_1^2 + m_2^2 + 2 * m_1 * q + q^2); \\ b_0 &= 1; \\ b_1 &= -q * (2 * m_0 * m_1 + m_1^2 + m_2^2 + (2 * m_0 + 4 * m_1) * q + 3 * q^2) / scale; \\ b_2 &= q^2 * (m_0 + 2 * m_1 + 3 * q) / scale; \\ b_3 &= -q^3 / scale; \\ B &= (m_0 * (m_1^2 + m_2^2) / scale)^2. \end{aligned} \quad (46)$$

Значение q вычисляется выражением

$$q = \begin{cases} -0,2568 + 0,5784 * \sigma + 0,0561 * \sigma^2, & \sigma < 3,556; \\ 2,5091 + 0,9804 * (\sigma - 3,556), & \sigma \geq 3,556, \end{cases} \quad (47)$$

где σ_0 – числовое значение параметра σ функции Габора.

Величины m_0, m_1, m_2 имеют следующие значения:

$$\begin{aligned} m_0 &= 1,16680; \\ m_1 &= 1,10783; \\ m_2 &= 1,40586. \end{aligned} \tag{48}$$

Для применения рекурсивных алгоритмов свертки с гауссианом и функцией Габора на изображениях необходимо выполнить прямую и обратную рекурсии по всем строкам и столбцам матрицы изображения. Главное преимущество данных алгоритмов состоит в их высокой скорости по сравнению с другими. Поскольку данные алгоритмы являются аппроксимационными, то они имеют погрешность.

Для вычисления максимальной абсолютной и квадратичной ошибки используются следующие выражения:

$$\begin{aligned} MAE(\sigma, \Omega) &= \max_n \{|\varepsilon(n|\sigma, \Omega)|\}; \\ SRE(\sigma, \Omega) &= \sqrt{\sum_n |\varepsilon(n|\sigma, \Omega)|^2}, \end{aligned} \tag{49}$$

где $\varepsilon(n|\sigma, \Omega)$ – комплексная разница между идеальным импульсным откликом $gabor(n|\sigma, \Omega)$ и результатом рекурсивной аппроксимации $out(n|\sigma, \Omega)$. Следовательно, $\varepsilon(n|\sigma, \Omega) = gabor(n|\sigma, \Omega) - out(n|\sigma, \Omega)$.

Поскольку $gabor(n|\sigma, \Omega) = gauss(n|\sigma)e^{j\Omega n}$ и $out(n|q, \Omega) = out_{gauss}(n|q) * e^{j\Omega n}$, где $out()$ и $out_{gauss}()$ являются рекурсивными аппроксимациями фильтров Габора и Гаусса, то можно записать

$$\begin{aligned} |\varepsilon(n|\sigma, \Omega)| &= |gabor(n|\sigma, \Omega) - out(n|q, \Omega)| = \\ &= |gauss(n|\sigma)e^{j\Omega n} - out_{gauss}(n|q)e^{j\Omega n}| = \\ &= |gauss(n|\sigma) - out_{gauss}(n|q)|. \end{aligned} \tag{50}$$

Главным недостатком рекурсивного алгоритма фильтрации функцией Габора является необходимость подстройки и ограничения параметров σ и ω для избежания эффекта наложения согласно формуле (22) из работы [28]: $1 \leq \sigma \leq \frac{N}{2\pi}$.

Для ускорения фильтрации гауссианом, а также для анализа изображений функциями Габора желательно использовать данные алгоритмы.

Заключение

В настоящей работе при рассмотрении алгоритмов быстрого вычисления признаков разных типов на цифровых изображениях было уделено внимание задачам вычисления комбинированных и антропометрических признаков лица, которые полностью не решены.

При использовании наборов фильтров Габора возникла задача снижения времени вычисления дискретной свертки. Для решения данной задачи авторы считают целесообразным применение фильтров Габора круглого

вида, а для решения задачи ускорения фильтрации гауссианом и функцией Габора – использование рекурсивных алгоритмов.

Для ускорения дискретной свертки были использованы четыре быстрых сверточных алгоритма. Представлен новый алгоритм быстрого вычисления дискретной свертки. При анализе алгоритма быстрого предварительного обнаружения объектов возникли задачи исследования и разработки новых типов слабых классификаторов алгоритма устойчивого обнаружения признаков, а также исследования влияния размера обучающего множества и количества слабых классификаторов на качество обнаружения объектов.

Литература

1. Анисимов Б., Курганов В., Злобин В. Распознавание и цифровая обработка изображений. – М.: Высшая школа, 1983. – 295 с.
2. Umbaugh S. Computer vision and image processing: A practical approach using CVPtools. – Prentice Hall, 1999. – 528 p.
3. Bow S. Pattern recognition and image processing. – Dekker, 2002. – 719 p.
4. Jain A. Fundamentals of digital image processing. – Prentice Hall, 1989. – 570 p.
5. Castleman R. Digital image processing. – Prentice Hall, 1996. – 667 p.
6. Фурман Я., Юрьев А., Яншин В. Цифровые методы обработки и распознавания бинарных изображений. – Красноярск: Изд-во Красноярского ун-та, 1992. – 248 с.
7. Haralick R., Shanmugam K., Dinstein I. Textural features for image classification // IEEE Transactions on Systems, Man and Cybernetics. – 1973. – Vol. 3. – № 11. – P. 610-621.
8. Старовойтов В.В. Локальные геометрические методы цифровой обработки и анализа изображений. – Мн: Ин-т техн. кибернетики НАН Беларуси, 1997. – 284 с.
9. Дюк В. Компьютерная психодиагностика. – М.: Братство, 1994. – 364 с.
10. Gabor D. Theory of communication // Journal of the IEE. – 1946. – Vol. 93 – P. 429-457.
11. Smith S., Brady J. SUSAN – A new approach to low level image processing // <http://portal.acm.org/citation.cfm?id=258056>
12. Претт У. Цифровая обработка изображений: в 2-х кн. – М.: Мир, 1982. – 712 с.
13. Фукунага К. Введение в статистическую теорию распознавания образов. – М.: Наука, 1979. – 368 с.
14. Penev P., Atick J. Local feature analysis: A general statistical theory for object representation // Computation in Neural Systems. – 1996. – Vol. 7. – № 3. – P. 477-500.

15. Hyvärinen A. Survey on independent component analysis // Neural Computing Surveys. – 1999. – P. 94-128.
16. Hyvrinen A., Oja E. A fast fixed-point algorithm for independent component analysis // Neural Computation. – 1997. – № 9. – P. 1483-1492.
17. Bell A., Sejnowski T. An information maximization approach to blind separation and blind deconvolution // Neural Computation. – 1995. – № 7. – P. 1129-1159.
18. Cardoso J. Higher order contrasts for independent component analysis // Neural Computation. – 1999. – № 11. – P. 157-192.
19. Loupías E., Sebe N. Wavelet-based salient points for image retrieval // http://rfv.insa-lyon.fr/~loupías/points/salientPoints_RR9911.pdf
20. Lee J., Sun Y., Chen C. Multiscale corner detection by using wavelet transform // IEEE Transactiona on Image Processing. – 1995. – Vol. 4. – № 1. – P. 100-104.
21. Chen C., Lee J., Sun Y. Wavelet transformation for gray-level corner detection // Pattern Recognition. – 1995. – Vol. 28. – № 6. – P. 853-861.
22. Bres S., Jolion J. Detection of interest points for image indexation // 3rd Int. Conf. on Visual Information Systems. – Amsterdam. – 1999. – P. 427-434.
23. Viola P., Jones M. Robust real-time object detection // Int. Journal of Computer Vision. – 2004. – Vol. 57. – № 2. – P. 137-154.
24. Шлезингер М. Алгоритм быстрой свертки // Теоретические и прикладные вопросы распознавания изображений: сб. науч. тр. – 1991. – С. 61-69.
25. Schapire R. A brief introduction to boosting // Sixteenth Int. Joint Conf. on Artificial Intelligence. – Stockholm. – 1999.
26. Schapire R., Freund Y., Barlett P., Lee W. Boosting the margin: a new explanation for the effectiveness of voting methods // The Annals of Statistics. – 1998. – Vol. 26. – P. 1651-1686.
27. Young I., Vliet L. Recursive implementation of Gaussian filter // Signal Processing. – 1999. – P. 139-151.
28. Young I., Vliet L., Ginkel M. Recursive Gabor filtering // IEEE Transaction on Signal Processing. – 2002. – Vol. 50. – P. 2798-2805.
29. Gotsman C. Constant time filtering with singular value decomposition // Computer Graphics Forum: special issue on Rendering. – 1994. – Vol. 16. – № 2. – P. 153-163.
30. Numerical Recipes in C / W. Press, S. Teukolsky, W. Vetterling, B. Flannery – Cambridge: Cambridge University Press. – 1992. – 994 p.
31. Dunn D., Higgins W., Wakeley J. Texture segmentation using 2-D Gabor elementary functions // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1994. – Vol. 16. – № 2. – P. 130-149.
32. Marcelija S. Mathematical description of the responses of simple cortical cells // Journal of the Optical Society of America. – 1980. – Vol. 70. – P. 1297-1300.
33. Daugman J. Two-dimensional spectral analysis of cortical receptive

field profile // Vision Research. – 1980. – Vol. 20. – P. 847-856.

34. Daugman J. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters // Journal of the Optical Society of America. – 1985. – Vol. 2. – № 7. – P. 1160-1169.

35. Liu C., Weshler H. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition // IEEE Transaction on Image Processing. – 2002. – Vol. 11. – № 4.

36. Discriminant gaborfaces and support vector machines classifier for face recognition / B. Zhang, W. Gao, S. Shan, Y. Peng // Asian Conf. on Computer Vision. – Korea. – 2004. – P. 37-42.

37. Lienhart R., Kuranov A., Pisarevsky V. Empirical analysis of detection cascades of boosted classifiers for rapid object detection // Proc. of the German 25th Pattern Recognition Symposium, Germany. – 2003.

38. Самаль Д.И. Алгоритмы идентификации человека по фотопортрету на основе геометрических преобразований: Автореф. дис. ... канд. техн. наук: 05.13.01. – Минск, 2002. – 20 с.

39. Freeman W., Adelson E. The desing and use of steerable filters // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1991. – Vol. 13. – № 9. – P. 891-906.

40. Sirovich L., Kirby M. Low-dimensional procedure for the characterization of human faces // Journal of the Optical Society of America. – 1987. – № 4. – P. 519-524.

41. Fournier A., Fiume E. Constant-time filtering with space-variant kernels // Proc. of the 15th annual Conf. on Computer Graphics and Interactive Techniques. – USA. – 1988. – P. 229-238.

42. Самаль Д.И., Старовойтов В.В. Подходы и методы распознавания людей по фотопортретам. – Минск, 1998. – 54 с. (Препринт/ Ин-т техн. кибернетики НАН Беларуси; № 8).

43. Брилюк Д.В., Старовойтов В.В. Распознавание человека по изображению лица нейросетевыми методами. – Минск, 2002. – 52 с. (Препринт/ Ин-т техн. кибернетики НАН Беларуси; № 2).

44. Baker S. Design and evaluation of feature detectors: PhD thesis. – Columbia University, 1998 // www.cs.columbia.edu/CAVE/publinks/baker_CUD_1998.pdf

45. Талёб М. Комбинированные алгоритмы сегментации цветных изображений: Автореф. дис. ... канд. техн. наук: 05.13.01. – Минск, 2002. – 20 с.

46. Haralick R., Shapiro L. Computer and Robot Vision. – Addison-Wesley, 1992. – 671 p.

47. Гуревич И. Проблема распознавания изображений // Распознавание, классификация, прогноз. – 1989. – Вып. 1. – С. 280-329.

48. Бутаков Е., Островский В., Фадеев И. Обработка изображений на ЭВМ. – М.: Радио и связь, 1987. – 240 с.

49. Kawato S. Circle-frequency filter and its application // Int. Workshop on Advanced Image Technology, ATR Media Integration and Communications

Research Laboratories. – 2001. – P. 217-222.

50. Jain A., Ross A., Prabhakar S. An introduction to biometric recognition // IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics. – 2004. – Vol. 14. – № 1. – P. 4-20.

51. Kawato S., Ohya J. Two-step approach for real-time eye tracking with a new filtering technique // IEEE Int. Conf. on Systems, Man and Cybernetics. – USA. – 2000. – P. 1366-1371.

52. Geusebroek J., Smeulders A., Weijer J. Fast anisotropic gauss filtering // IEEE Transactions on Image Processing. – 2003. – Vol. 12. – № 8. – P. 938-943.

53. Liu C. Gabor-based kernel PCA with fractional power polynomial models for face recognition // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2004. – Vol. 26. – № 5. – P. 572-581.

54. McNames J. A fast nearest-neighbor algorithm based on a principal axis search tree // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2001. – Vol. 23. – № 9. – P. 964-976.

55. Gil J., Kimmel R. Efficient dilation, erosion, opening, and closing algorithms // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2002. – Vol. 24. – P. 1606-1617.

56. Feris R., Gemmell J., Toyama K., Kruger V. Facial feature detection using a hierarchical wavelet face database // <http://research.microsoft.com/~JGemmell/pubs/WaveBaseTR-2002-05.pdf>

57. Heckbert P. Survey of texture mapping // IEEE Computer Graphics and Applications. – 1986. – P. 56-67.

58. Lowe D. Object recognition from local scale-invariant features // Proc. of the Int. Conf. on Computer Vision. – Greece. – 1999. – P. 1150-1157.

59. Martin D., Fowlkes C., Malik J. Learning to detect natural image boundaries using local brightness, color, and texture cues // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2004. – Vol. 26. – № 5. – P. 530-549.

Содержание

Введение	3
1 Признаки объектов, вычисляемые на цифровых изображениях	4
1.1 Геометрические признаки	4
1.1.1 <i>Особые точки</i>	5
1.1.2 <i>Границы объекта</i>	5
1.1.3 <i>Форма объекта</i>	5
1.1.4 <i>Размеры объекта</i>	5

1.2	Яркостные признаки	6
1.2.1	Экстремумы	6
1.2.2	Частотные признаки	6
1.2.3	Гистограммные признаки	6
1.2.4	Моменты	6
1.3	Текстурные признаки	7
1.4	Бинарные признаки	9
1.5	Комбинированные признаки	9
2	Классификация алгоритмов вычисления признаков	9
2.1	Инвариантность признаков	9
2.2	Алгоритмы вычисления признаков	10
3	Алгоритмы предварительного обнаружения объектов	12
3.1	Обнаружение объектов с использованием интегральных матриц изображений и алгоритма AdaBoost	12
3.2	Использование детектора SUSAN для вычисления признаков объектов	18
4	Алгоритмы ускорения вычисления дискретной свертки	21
4.1	Интегрально-дифференциальный алгоритм вычисления свертки	22
4.2	Сингулярное разложение фильтра для ускорения свертки	25
4.3	Алгоритм сингулярного разложения и разрежения векторов	27
4.4	Экспериментальный анализ алгоритмов свертки	28
5	Фильтры Габора для вычисления яркостных и текстурных признаков	29
5.1	Использование фильтров Габора для анализа цифровых изображений	29
5.2	Минимизация набора фильтров Габора	30
6	Ускорение вычисления свертки через рекурсию	32
6.1	Рекурсивный алгоритм свертки с гауссианом	32
6.2	Рекурсивный алгоритм свертки с функцией Габора	33
	Заключение	34
	Литература	35