

УДК 004.932, 004.272

СИНТЕЗ МОДЕЛИ ПЕРЕСТРАИВАЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ ДЛЯ ОПРЕДЕЛЕНИЯ ХАРАКТЕРИСТИК ГРАДИЕНТА ИЗОБРАЖЕНИЯ

© А. С. Бондарчук, Д. В. Шашев, С. В. Шидловский

*Томский государственный университет,
634050, г. Томск, просп. Ленина, 36
E-mail: shidlovskiyv@mail.ru*

Рассматривается новый подход к вычислению характеристик градиента полутонового изображения как массива признаков объекта интереса. Предлагается синтезировать модель перестраиваемой вычислительной среды, которая может одновременно в параллельном режиме обработать каждый пиксель исходного изображения и сформировать массив с характеристиками градиента. Благодаря архитектурным принципам построения модели расчёт градиента выполняется за 1 такт элементарного вычислителя перестраиваемой вычислительной среды.

Ключевые слова: полутоновое изображение, вычисление градиента, перестраиваемая вычислительная среда.

DOI: 10.15372/AUT20210203

Введение. За последнее время количество автономных транспортных средств значительно выросло. Именно системы технического зрения (СТЗ) позволяют автономным транспортным средствам реагировать на внешнюю среду и преодолевать различные препятствия. Наряду с развитием технологий создания устройств приёма и передачи визуальных данных этому также способствуют многочисленные исследования в области анализа и обработки цифровых сигналов применительно к различным задачам, а последние 10 лет акцент исследований смещён в область построения алгоритмов машинного обучения [1–7]. Обеспечение беспилотных автомобилей способностью реагировать на окружающую среду и ориентироваться в пространстве в условиях городской среды является актуальной задачей современности.

Системы технического зрения, использующие методы машинного обучения, позволяют понимать окружающую среду путём анализа видеоданных. С точки зрения использования СТЗ в системах управления беспилотными автомобилями одной из ключевых задач является обнаружение пешеходов. Быстрое и своевременное детектирование пешеходов даёт возможность реагировать на них и избегать столкновений, гарантируя при этом безопасность дорожного движения. Соответственно при решении данной задачи необходимо обеспечить максимальную скорость детектирования, так как от неё зависит время реакции системы управления беспилотным автомобилем. Чтобы уменьшить требования к вычислительной системе и повысить её быстродействие, нужны новые подходы в области построения алгоритмов распознавания и классификации объектов на изображениях.

Одним из таких подходов является реализация алгоритмов, выполняемых на вычислительных архитектурах параллельно-конвейерного типа, среди которых наиболее часто используются процессоры Field-Programmable Gate Arrays (FPGA) и Graphic Processing Unit (GPU) либо гетерогенные вычислительные системы совместно с многопроцессорными Central Processing Unit (CPU) [8–10].

В свою очередь, в данной работе предлагается альтернативный подход, при котором алгоритмы распознавания и классификации объектов на изображении подвергаются модификации для их аппаратного выполнения в специализированных перестраиваемых вычислительных средах (ПВС), архитектурные особенности которых позволяют достичь многократного увеличения быстродействия выполнения данных алгоритмов.

Задача детектирования пешеходов в общем случае разделяется на два этапа: выделение признаков объектов интереса и классификация этих объектов. В настоящее время решение задачи сводится к применению дескриптора направленных градиентов (Histogram of Oriented Gradients, HOG) для выделения признаков пешеходов и их классификации с использованием алгоритма опорных векторов (Support Vector Machine, SVM) [11, 12].

В рамках представленной работы рассматриваются принцип и процесс создания модифицированного алгоритма HOG, реализуемого в модели ПВС для обработки полутонового изображения.

Принципы работы алгоритма HOG. Дескриптор — это описательный элемент изображения или области изображения, который извлекает полезную информацию и отбрасывает лишнюю. Дескрипторы необходимы для таких задач, как распознавание образов и обнаружение объектов на изображении. Вектор признаков является дескриптором и состоит из определённого числа последовательных значений, описывающих характерные черты изображения. Векторы признаков используются для классификации объектов на изображении с помощью, например, алгоритмов SVM.

HOG — это дескриптор особых точек, который основан на расчёте направлений градиента в локальных областях изображения [11]. Основная идея алгоритма — допущение, что внешний вид и форма объекта на изображении могут быть описаны распределением градиентов интенсивности. Значения градиентов рассчитываются в горизонтальном и/или вертикальном направлении с помощью одномерной дифференцирующей маски. Этот метод требует фильтрации цветовой или яркостной составляющей с помощью следующих фильтрующих ядер: $[-1, 0, 1]$ и $[-1, 0, 1]^T$.

Для того чтобы вычислить HOG, направление и величина градиента для каждого пикселя в изображении рассчитываются по формулам (1)–(4):

$$g_x = f(x + 1, y) - f(x - 1, y), \quad (1)$$

$$g_y = f(x, y + 1) - f(x, y - 1), \quad (2)$$

$$m(x, y) = \sqrt{g_x^2 + g_y^2}, \quad (3)$$

$$\theta(x, y) = \tan^{-1}(g_y/g_x). \quad (4)$$

Здесь $f(x, y)$ — значение яркости пикселя с координатами (x, y) ; $\theta(x, y)$ — направление градиента; $m(x, y)$ — величина градиента пикселя (x, y) .

Для пикселей на границе изображения недостаточно информации, чтобы вычислить значения величины и направления градиента, поэтому эти пиксели далее не используются. Градиенты отображают на изображении места, где происходит резкое изменение яркости пикселей. Величина градиентов больше на краях и углах объекта, которые содержат гораздо больше информации о форме объекта, чем однородные области. Таким образом, градиенты выделяют контуры объектов и отбрасывают ненужную информацию, такую как однородный фон.

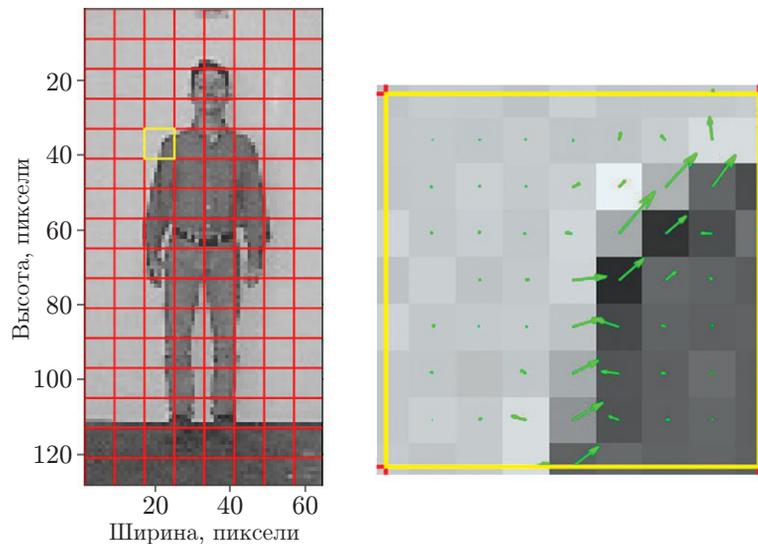


Рис. 1. Визуализация градиентов пикселей изображения

Направление градиента рассчитывается в радианах и принимает значения в диапазоне $(-\pi, \pi)$, что называется «градиент со знаком». Далее радианы переводят в градусы и получают значения в диапазоне $(0^\circ, 360^\circ)$. Тем не менее для повышения производительности градиенты лучше использовать без знака. Обнаружение объекта на изображении основано главным образом на обнаружении краёв и не требует знания знака градиента, поскольку цвета пикселей не так важны. Например, направление градиента, равное 90° , и противоположное ему направление в 270° считаются одинаковыми. Не столько важно, куда именно направлен градиент, сколько важна общая тенденция изменения интенсивности пикселей, для того чтобы определить границу объектов. Другими словами, градиенты должны быть преобразованы таким образом, чтобы их значения были в диапазоне $(0^\circ, 180^\circ)$. Решение такого преобразования состоит в том, чтобы прибавить число π к отрицательным направлениям.

После нахождения величины и направления градиента для каждого пикселя изображение делится на небольшие пространственные области, которые называются ячейками. Для примера изображение было разделено на ячейки размером 8×8 пикселей. На рис. 1 продемонстрирована одна ячейка изображения, на которую наложены векторы НОГ. Стрелки указывают направление градиента, а их длины зависят от его величины.

Для каждой ячейки рассчитывается гистограмма направлений градиента, в которой каждый бин представляет собой сумму величин градиентов внутри ячейки на определённом интервале направлений. Гистограмма состоит из 9 бинов, где каждый бин является определённым углом направления градиента. Диапазон гистограммы составляет от 0° до 180° , при этом диапазон каждого отдельного бина равен 20° . Распределение величин по бинам гистограммы происходит следующим образом. Сначала для каждого пикселя определяются бины (два или один), в которые будут вноситься значения. Затем значения рассчитываются с использованием линейной интерполяции относительно центра этих бинов. Значения, вносимые в бин, суммируются для создания гистограммы. Если угол направления градиента больше 170° , то его величина разделяется пропорционально между бинами 170° и 10° . Для угла $177,33^\circ$ и величины 43,04 значение, вносимое в бин 10° , рассчитывается следующим образом:

$$((177,33^\circ - 170^\circ) \cdot 43,04) / 20^\circ = 15,77.$$

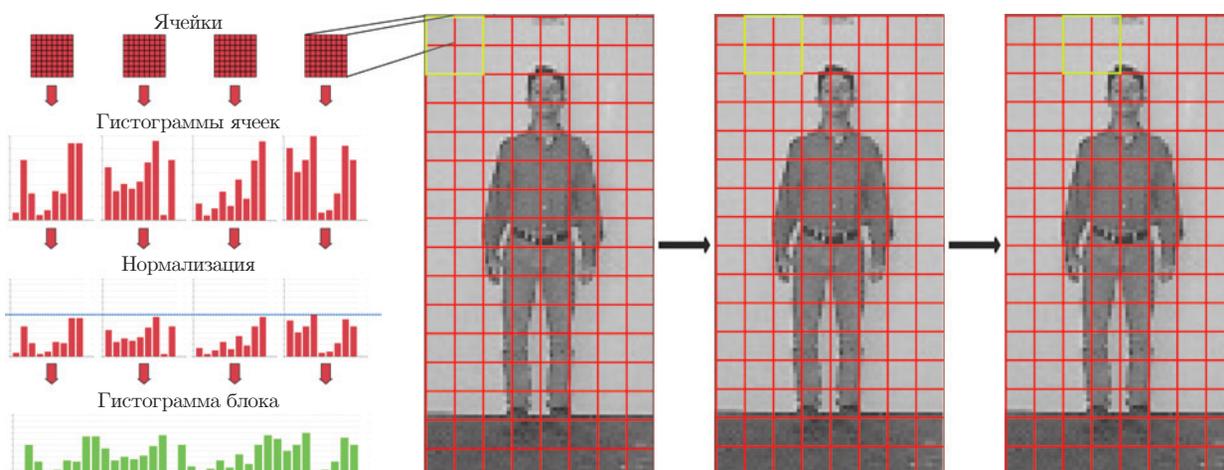


Рис. 2. Формирование гистограммы блока

Значение, вносимое в бин 170° , рассчитывается по формуле

$$43,04 - 15,77 = 27,27.$$

После построения гистограмм ячейки объединяются в блоки. Блоки представляют собой пересекающиеся области на изображении, которые содержат несколько ячеек. Для примера возьмём блок размером 2×2 ячейки. Блок используется для определения ячеек, гистограммы которых должны быть нормализованы вместе. Смысл этого в том, чтобы сделать конечный вектор признаков устойчивым к изменениям значений пикселей, вызванным, например, изменениями освещённости. На рис. 2 изображена схема построения гистограммы блока.

Поскольку в каждом блоке находится четыре ячейки, он объединяет четыре гистограммы. Гистограмма блока состоит из 36 бинов, значения которых нормализованы $L2$ -нормой. Математическое представление $L2$ -нормы:

$$H_{L2} = \frac{H}{\sqrt{H_1^2 + H_2^2 + \dots + H_n^2}},$$

где H_{L2} — нормализованная гистограмма, H — гистограмма блока, n — количество бинов в гистограмме, H_n — значение n -го бина гистограммы.

Окончательно нормализованные гистограммы всех блоков объединяются в одну большую гистограмму, характеризующую всё изображение. Для изображения размером 64×128 пикселей количество блоков размером 2×2 ячейки равно 105. Размер гистограммы одного блока равен 36 бином. Следовательно, размер гистограммы всего изображения будет равен 3780 бином. Сформированная гистограмма представляет собой вектор признаков, который в дальнейшем подаётся в алгоритм классификации.

Перестраиваемая вычислительная среда — дискретная математическая модель высокопроизводительной вычислительной системы, состоящей из одинаковых и одинаково соединённых друг с другом простейших универсальных элементов (элементарных вычислителей (ЭВ) / многофункциональных логических модулей), программно настраиваемых на выполнение любой функции из заданного класса, памяти и любого соединения со своими соседями [13].

Результаты многочисленных исследований обобщены, и на их базе сформулированы основные принципы создания ПВС [13–16]: параллельность, перестраиваемость, однородность, конвейерность обработки информации.

Отметим, что ПВС представляет собой частный случай матричного процессора [17] с действующими ограничениями на вычислительную архитектуру:

— ПВС представляет собой геометрически правильную решётку, имеющую не менее двух осей симметрии, в узлах которой расположены ЭВ со строго определённым набором выполняемых операций;

— каждый ЭВ имеет настроечный вход, с помощью которого можно осуществлять его перестраиваемость на выполнение одной из закладываемых операций. Он может быть настроен на выполнение в данный момент любой одной функции хотя бы одного полного базиса и функции канала передачи сигналов в заданном направлении, т. е. обладает функциональной и соединительной полнотой;

— все ЭВ однотипны и геометрически одинаковым образом соединены с соседними, причём каждый из ЭВ можно условно считать центром симметрии относительно связей его с окружающими ЭВ;

— выполняемые операции в ЭВ и соединения между ними должны обеспечить аппаратное выполнение реализуемого алгоритма.

Перестраиваемую вычислительную среду также можно дополнить понятием перестраиваемого автомата и рассмотреть её работу с точки зрения теории автоматов. Перестраиваемым называется автомат, для которого задано множество реализуемых им автоматных отображений и определён алгоритм настройки на реализацию каждого из автоматных отображений [13]. Автоматное отображение — это однозначное отображение зависимости вектора выходов автомата от вектора входов, а настройка на реализацию каждого из автоматных отображений осуществляется путём определения их кодов настройки. Следует отметить, что в данной работе рассматриваются автоматы без памяти или комбинационные схемы.

Синтез модели ПВС для вычисления характеристик градиента изображения. С точки зрения работы алгоритма НОГ наиболее важным этапом является извлечение характеристик градиента для каждого пикселя изображения. Дальнейшие преобразования полученных данных о величине градиента и его направлении относятся, скорее, к этапу их нормализации и оптимизации, а также сокращению объёма для более эффективной работы классификатора для производительности.

В работе предлагается технология извлечения характеристик градиента, при которой каждый ЭВ ПВС отвечает за обработку одного из пикселей исходного изображения. Таким образом, размеры ПВС и изображения совпадают, а характеристики градиента вычисляются параллельно для каждого пикселя.

Согласно формулам (1)–(4), чтобы вычислить характеристики градиента для рассматриваемого пикселя, необходимо использовать информацию об интенсивностях пикселей соседей, расположенных в направлениях север, юг для определения градиента по вертикали и запад, восток для определения градиента по горизонтали. Соответственно синтезируемая архитектура модели ПВС должна обеспечивать 4-связный тип соединения между ЭВ.

Для синтеза модели ПВС зададим автоматные отображения, реализующие вычисление характеристик градиента (рис. 3). Присвоим каждому автоматному отображению код настройки (z_4, z_3, z_2, z_1) , при котором автомат перестраивается на него, а также системы выходных формул, т. е. запишем автоматные операторы:

1) при $z_4 = 0, z_3 = 0, z_2 = 0, z_1 = 0$ (рис. 3, а)

$$f_2 = f_4 = m = \theta = 0, \quad f_1 = f_3 = x;$$

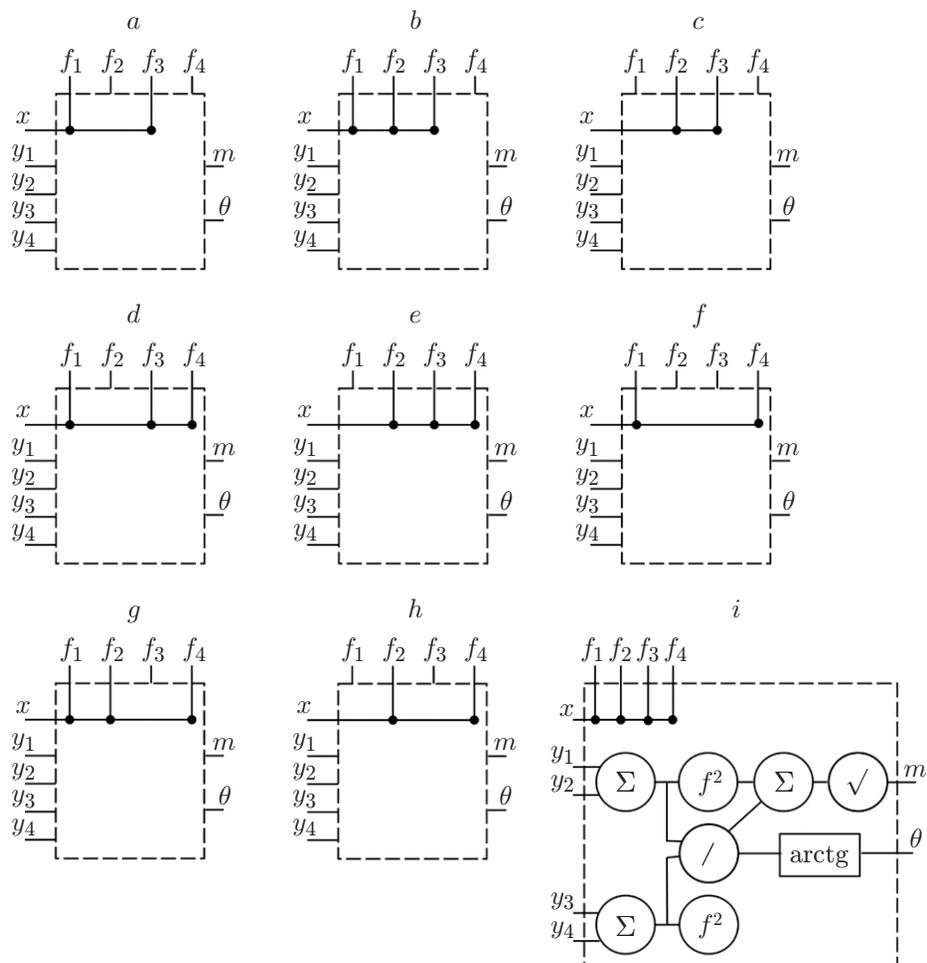


Рис. 3. Автоматные отображения (x, y_i — информационные входы, m, θ — выходы автомата, f_j — выходы межавтоматных связей): $a - i$ — отображения при различных кодах настройки ($z_1 - z_4$)

2) при $z_4 = 1, z_3 = 0, z_2 = 0, z_1 = 0$ (рис. 3, b)

$$f_4 = m = \theta = 0, \quad f_1 = f_2 = f_3 = x;$$

3) при $z_4 = 0, z_3 = 1, z_2 = 0, z_1 = 0$ (рис. 3, c)

$$f_1 = f_4 = m = \theta = 0, \quad f_2 = f_3 = x;$$

4) при $z_4 = 1, z_3 = 1, z_2 = 0, z_1 = 0$ (рис. 3, d)

$$f_2 = m = \theta = 0, \quad f_1 = f_3 = f_4 = x;$$

5) при $z_4 = 0, z_3 = 0, z_2 = 1, z_1 = 0$ (рис. 3, e)

$$f_1 = m = \theta = 0, \quad f_2 = f_3 = f_4 = x;$$

6) при $z_4 = 1, z_3 = 0, z_2 = 1, z_1 = 0$ (рис. 3, f)

$$f_2 = f_3 = m = \theta = 0, \quad f_1 = f_4 = x;$$

7) при $z_4 = 0, z_3 = 1, z_2 = 1, z_1 = 0$ (рис. 3, *g*)

$$f_3 = m = \theta = 0, \quad f_1 = f_2 = f_4 = x;$$

8) при $z_4 = 1, z_3 = 1, z_2 = 1, z_1 = 0$ (рис. 3, *h*)

$$f_1 = f_3 = m = \theta = 0, \quad f_2 = f_4 = x;$$

9) при $z_4 = 0, z_3 = 0, z_2 = 0, z_1 = 1$ (рис. 3, *i*)

$$f_1 = f_2 = f_3 = f_4 = x, \quad m = \sqrt{(y_1 - y_2)^2 + (y_3 - y_4)^2}, \quad \theta = \operatorname{arctg} \left(\frac{y_1 - y_2}{y_3 - y_4} \right).$$

На базе структурно-автоматного метода [17] получаем следующую систему формул для описания работы ЭВ ПВС:

$$f_1 = x(z_4 \bar{z}_1 \bar{z}_2 + z_4 \bar{z}_1 \bar{z}_3 + \bar{z}_2 \bar{z}_3 \bar{z}_4 + z_2 z_3 \bar{z}_1 \bar{z}_4);$$

$$f_2 = x(z_2 z_3 \bar{z}_1 + z_2 \bar{z}_1 \bar{z}_4 + z_3 \bar{z}_1 \bar{z}_4 + z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4 + z_4 \bar{z}_1 \bar{z}_2 \bar{z}_3);$$

$$f_3 = x(\bar{z}_1 \bar{z}_2 + \bar{z}_1 \bar{z}_3 \bar{z}_4 + \bar{z}_2 \bar{z}_3 \bar{z}_4); \quad f_4 = x(z_2 \bar{z}_1 + z_3 z_4 \bar{z}_1 + z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4); \quad (5)$$

$$m = \sqrt{(y_1 - y_2)^2 + (y_3 - y_4)^2} \quad z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4; \quad \theta = \operatorname{atg} \left(\frac{y_1 - y_2}{y_3 - y_4} \right) z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4.$$

На рис. 3 автоматные отображения *a–h* описывают транспортные функции ЭВ по передаче информации соседним ЭВ. Выполнение данных автоматных отображений необходимо для того, чтобы учесть в обработке исходного изображения краевые пиксели, у которых отсутствуют соседние пиксели, что указывает на невозможность расчёта характеристик градиента по формулам (1)–(4).

В описании автоматного отображения *i* отражены классические алгебраические операции суммирования, возведения в квадрат, деления, вычисления квадратного корня от числа, а также вычисления арктангенса в соответствии с формулами (1)–(4).

Таким образом, ЭВ синтезируемой модели ПВС содержит в себе совокупность описанных выше автоматных отображений и способен перестраиваться на их выполнение с помощью настроечного кода (z_4, z_3, z_2, z_1) , т. е. выполняет систему уравнений (5).

Архитектура модели синтезируемой ПВС размером 3×3 ЭВ представлена на рис. 4. Данная модель отражает обработку верхнего левого угла исходного изображения размером 3×3 пикселя. Здесь X — массив пикселей исходного изображения; Z — массив настроечных кодов для ЭВ, причём на рисунке уже отмечены правильные коды в соответствии с автоматными отображениями; m и θ — массивы величин и направлений градиентов соответственно. Под термином «шина данных» здесь необходимо понимать прямой канал передачи цифровых данных, а многоканальность шины введена для удобства отображения нескольких однотипных каналов одной линией.

Таким образом, в модели ПВС осуществляются одновременная параллельная попиксельная обработка изображения и вычисление характеристик градиента для каждого пикселя.

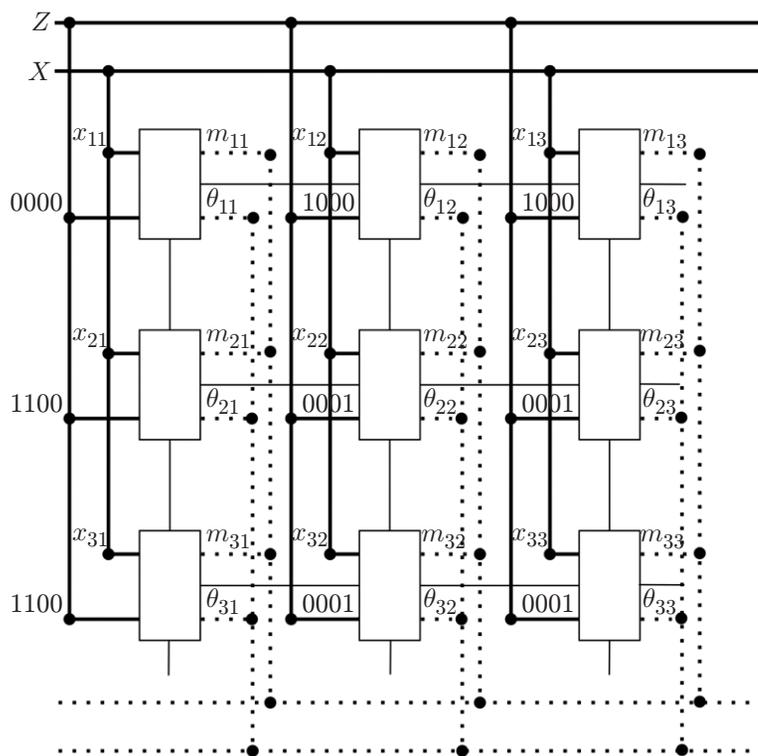


Рис. 4. Архитектура перестраиваемых вычислительных сред: Z (сплошная полужирная линия) — 4-канальная шина входных настроечных данных; X (сплошная полужирная линия) — шина входных данных с изображения; пунктирная линия — шина выходных данных; сплошная тонкая линия — 2-канальная шина связи, обеспечивающая обмен данными между входами и выходами f каждого ЭВ

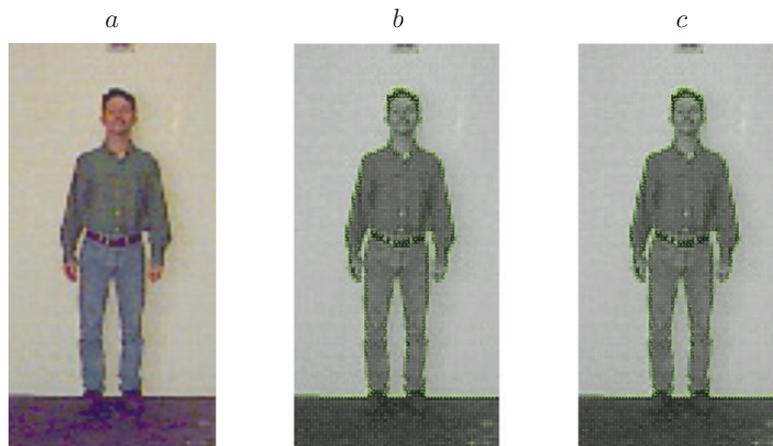


Рис. 5. Результаты обработки тестового изображения: a — исходное изображение, b — после обработки синтезированной ПВС, c — после применения классического НОГ

Результат моделирования. Моделирование разработанного алгоритма вычисления характеристик градиента с использованием ПВС, а также проверка корректности работы данного алгоритма проводились в MATLAB Simulink путём сравнения результатов работы ПВС и классического модуля HOG, доступного в Computer Vision Toolbox в среде MATLAB.

Результат работы синтезированной модели ПВС и классического HOG представлен на рис. 5. Для наглядной оценки вычисления характеристик градиента на исходное изображение был наложен массив градиентов в виде векторов для каждого пикселя.

Изображения, полученные при использовании ПВС (рис. 5, *b*) и в MATLAB (рис. 5, *c*), идентичны, что свидетельствует о корректности работы ПВС.

Заключение. В представленной работе приведены результаты синтеза модели ПВС для вычисления характеристик градиента каждого пикселя исходного изображения. Полученные данные о градиенте в дальнейшем используются для классификации и распознавания объектов на изображении. Показано, что с помощью ПВС можно добиться таких же результатов, что и при использовании классического алгоритма HOG, однако время обработки в первом случае значительно сокращается. Согласно представленной архитектуре синтезированной модели ПВС (см. рис. 4) вычисление массива с характеристиками градиента для каждого пикселя изображения выполняется параллельно и за 1 такт работы ЭВ ПВС.

Исследование области применения ПВС для решения задач компьютерного зрения является, по мнению авторов, перспективным направлением, так как данный подход позволяет реализовать вычислительно ёмкие алгоритмы обнаружения и распознавания объектов на изображении с высоким показателем быстродействия, что крайне необходимо в построении современных систем технического зрения для беспилотного транспорта. Одна из возможных практических реализаций моделей ПВС заключается в формировании синтезируемых архитектур в программируемых логических интегральных схемах FPGA, чему и будут посвящены дальнейшие исследования. Аналогичный подход к построению реконфигурируемых полупроводниковых фотодиодных матриц для реализации нейросетевых алгоритмов был представлен в работе [18].

Финансирование. Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 19-37-90110).

СПИСОК ЛИТЕРАТУРЫ

1. **Chen J., Ran X.** Deep learning with edge computing: A review // Proc. of the IEEE. 2019. **107**, N 8. P. 1655–1674.
2. **Arnold E., Al-Jarrah O., Dianati M. et al.** A survey on 3D object detection methods for autonomous driving applications // IEEE Trans. Intell. Transportation Syst. 2019. **20**, N 10. P. 3782–3795.
3. **Li D., Zhao D., Zhang Q., Chen Y.** Reinforcement learning and deep learning based lateral control for autonomous driving [Application Notes] // IEEE Comput. Intell. Magazine. 2019. **14**, N 2. P. 83–98.
4. **You C., Lu J., Filev D., Tsiotras P.** Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning // Robotics and Autonomous Syst. 2019. **114**. P. 1–18.
5. **Luong N. C., Hoang D. T., Gong S. et al.** Applications of deep reinforcement learning in communications and networking: A survey // IEEE Communications Surveys & Tutorials. 2019. **21**, N 4. P. 3133–3174.

6. **Сурин В. А., Тырсин А. Н.** Модель нелинейного фильтра для цифровой обработки контрастных изображений // *Автометрия*. 2018. **54**, № 2. С. 54–62. DOI: 10.15372/AUT20180206.
7. **Першина Ж. С., Каздорф С. Я., Лопота А. В.** Методы визуальной навигации мобильного робота и построения картографических моделей внешней среды // *Автометрия*. 2019. **55**, № 2. С. 92–102. DOI: 10.15372/AUT20190210.
8. **Feng X., Jiang Y., Yang X. et al.** Computer vision algorithms and hardware implementations: A survey // *Integration*. 2019. **69**. P. 309–320.
9. **Nakahara H., Yonekawa H., Fujii T., Sato S.** A lightweight YOLOv2: A binarized CNN with a parallel support vector regression for an FPGA // *Proc. of the ACM/SIGDA Intern. Symposium on Field-Programmable Gate Arrays*. Monterey, USA, 25–27 Febr., 2018. P. 31–40.
10. **Лысаков К. Ф., Облаухов К. К., Шадрин М. Ю.** Реализация на базе FPGA алгоритмов выявления искажений изображения в результате компрессии // *Автометрия*. 2020. **56**, № 1. С. 33–38. DOI: 10.15372/AUT20200104.
11. **Dalal N., Triggs B.** Histograms of oriented gradients for human detection // *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, USA: IEEE, 2005. P. 886–893.
12. **Luo J. H., Lin C. H., Li X.** Pure FPGA implementation of an HOG based real-time pedestrian detection system // *Sensors*. 2018. **18**, N 4. 1174.
13. **Шидловский С. В.** Автоматическое управление. Перестраиваемые структуры. Томск: Томский гос. университет, 2006. 288 с.
14. **Каляев И. А., Левин И. И., Семерников Е. А., Шмойлов В. И.** Реконфигурируемые мультимедийные вычислительные структуры. Ростов н/Д: Изд-во ЮНЦ РАН, 2008. 393 с.
15. **Хорошевский В. Г., Курносоев М. Г., Мамойленко С. Н.** Пространственно-распределённая мультимедийная вычислительная система: архитектура и программное обеспечение // *Вестн. ТГУ. Управление, вычислительная техника и информатика*. 2011. № 1(14). С. 79–84.
16. **Шашев Д. В., Шидловский С. В.** Морфологическая обработка бинарных изображений с использованием перестраиваемых вычислительных сред // *Автометрия*. 2015. **51**, № 3. С. 19–26.
17. **Шидловский С. В.** Автоматическое управление. Перестраиваемые структуры в системах с распределёнными параметрами. Томск: Томский гос. университет, 2007. 192 с.
18. **Mennel L., Symonowicz J., Wachter S. et al.** Ultrafast machine vision with 2D material neural network image sensors // *Nature*. 2020. **579**. P. 62–66.

Поступила в редакцию 11.08.2020

После доработки 23.10.2020

Принята к публикации 18.01.2021