

УДК 004.932

СЕГМЕНТАЦИЯ ПАРКОВОЧНЫХ МЕСТ НА ИЗОБРАЖЕНИЯХ АВТОСТОЯНОК

*П.В. ЯРОШЕВИЧ, канд. техн. наук, доц. Р.П. БОГУШ,
канд. техн. наук, доц. Д.О. ГЛУХОВ
(Полоцкий государственный университет)*

Предложен алгоритм сегментации парковочных мест, который включает преобразование перспективы исходного изображения, адаптивную бинаризацию методом Отсу, применение операций математической морфологии, горизонтальное дифференцирование изображения, построение вертикальных линий, фильтрацию и определение координат парковочных мест. Для проведения экспериментов выполнена программная реализация алгоритма. Рассмотрены реализованные классы и представлены результаты выполненных экспериментальных исследований.

Ключевые слова: машинное зрение, анализ изображений, структурирование мест парковки.

Введение. В последние годы благодаря развитию технологий компьютерного зрения предложен ряд видеосистем с интеллектуальной обработкой входных видеоданных для решения различных прикладных задач, включая системы управления автомобильными парковками [1–3]. Одним из этапов алгоритмического обеспечения таких систем является сегментация и локализация парковочных мест на изображениях. Общий подход при построении алгоритмов таких систем предполагает автоматическое обнаружение свободных и занятых парковочных мест. Для этого необходимо выполнение следующих основных этапов: предварительная сегментация и локализация парковочных мест (регионов интереса), вычисление их признаков, классификация полученных дескрипторов и интерпретация результата. Точное определение расположения парковочного места на изображении необходимо при выполнении ряда задач, например, при установке нового программно-аппаратного комплекса управления парковкой, при переносе точки обзора, при модернизации оборудования либо замене программного обеспечения и т.д.

В работе [4] для анализа занятости парковки используются изображения, полученные на основе аэросъемки. При этом предлагаемый алгоритм учитывает особенность размещения автомобилей над поверхностью Земли и построен на основе анализа карт интенсивности и высот для сегментации автомобиля на парковочном месте. После сегментации парковочных мест и автомобилей на них, восстанавливается изображение парковки со свободными парковочными местами. На основе изображений автомобилей генерируются 3D-структуры, которые затем используются для моделирования активности на парковке. Для эффективной работы метода требуется четкая видимость разделяющих полос. Таким образом, при достаточно большой занятости парковки алгоритм некорректно сегментирует регионы интереса и автомобили. В статье [5] рассматривается структурирование парковочных мест на автостоянке на основе комбинированного метода, включающего два шага. При этом первый шаг выполняет оператор, который выделяет границы каждого места парковки белым цветом и присваивает центральному пикселю желтый цвет. На втором шаге на основе программной реализации алгоритма выполняется автоматический поиск преобразованных пикселей желтого цвета и определение координат границ обнаруженного региона интереса. В [6] авторы предлагают алгоритм определения расположения парковочных мест, состоящий из следующих шагов: определение расположения линий, применение процедур фильтрации, предварительное объединение линий в парковочные места и ряды, операции интерполяции, экстраполяции и предсказания для обнаружения пропущенных мест парковки. Метод достаточно эффективен, однако необходимо использовать изображения, полученные на основе аэросъемки с четкой видимостью разделяющих полос.

В данной работе предлагается эффективный алгоритм для автоматической сегментации парковочных мест на изображениях автомобильных парковок, получаемых с использованием камер видеонаблюдения, который включает преобразование перспективы исходного изображения, адаптивную бинаризацию методом Отсу, применение операций математической морфологии, горизонтальное дифференцирование изображения, построение вертикальных линий, фильтрацию и определение координат парковочных мест.

1. Алгоритм сегментации парковочных мест. Предложенный алгоритм сегментации и локализации парковочных мест состоит из последовательности следующих шагов:

1. Преобразование перспективы изображения парковки. На исходном изображении определяются четыре исходные точки (x_i, y_i) $i = \overline{1,4}$, координаты которых на преобразованном изображении являются

вершинами прямоугольной области (x'_i, y'_i) $i = \overline{1,4}$. Данная область характеризуется минимальной площадью, которая охватывает исходные точки. Выполняется преобразование перспективы:

$$(x'_i, y'_i, 1) = H \times (x, y, 1)^T,$$

где H – матрица гомографии.

Матрица гомографии формируется на основе вектора $G_H = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})^T$ в результате решения системы линейных алгебраических уравнений:

$$V \cdot G_H = Y',$$

где V – матрица размера 8×8 :

$$V = \begin{bmatrix} \overline{(x, y, 1)}^T & 0 & \overline{-(x \cdot x')}^T & \overline{-(x' \cdot y')}^T \\ 0 & \overline{(x', y', 1)}^T & \overline{-(x \cdot y')}^T & \overline{-(y \cdot y')}^T \end{bmatrix},$$

Y' – вектор-столбец размера 8×1 :

$$Y' = (x'_1, x'_2, x'_3, x'_4, y'_1, y'_2, y'_3, y'_4)^T.$$

2. Бинаризация изображения методом Отсу. Данный алгоритм, используя пороговое значение, делит гистограмму изображения на два класса [7].

Полагая, что пиксели подразделяются на два класса, алгоритм минимизирует взвешенную дисперсию $\sigma_w^2(t)$ между классами, объявленную в выражении

$$\sigma_w^2(t) = q_1(t) \cdot \sigma_1^2(t) + q_2(t) \cdot \sigma_2^2(t),$$

где t – порог, $t \in 0 \dots 255$.

Функция вероятности P определяется для каждого значения пикселя. На первом шаге вычисляется гистограмма распределения для изображения, далее осуществляется нормализация для гарантирования вероятностного распределения. Затем значения пикселей разделяются на два класса C_1 и C_2 при использовании порога t и вероятностей $q_1(t)$ и $q_2(t)$ соответственно:

$$q_1(t) = \sum_{i=1}^t P(i), \quad q_2(t) = \sum_{i=t+1}^I P(i),$$

где значения пикселей двух классов C_1 и C_2 находятся в промежутках $[1, t]$ и $[t + 1, I]$ соответственно.

Тогда значения средних $\mu_1(t)$ и $\mu_2(t)$ для классов C_1 и C_2 вычисляются следующим образом:

$$\mu_1(t) = \sum_{i=1}^t \frac{i \cdot P(i)}{q_1(t)}, \quad \mu_2(t) = \sum_{i=1}^I \frac{i \cdot P(i)}{q_2(t)};$$

дисперсии для тех же классов:

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}, \quad \sigma_2^2(t) = \sum_{i=1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}.$$

Значение дисперсии показывает эффективность выбора пороговой величины. Полная дисперсия вычисляется следующим выражением:

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2(t),$$

где $\sigma_w^2(t)$ – функция уровня порога t , основанная на дисперсии;

$\sigma_b^2(t)$ – функции уровня порога t , основанная на математическом ожидании:

$$\sigma_b^2(t) = q_1(t) \cdot q_2(t) \cdot [\mu_1(t) - \mu_2(t)]^2.$$

Алгоритм направлен на уменьшение $\sigma_w^2(t)$ или увеличение $\sigma_b^2(t)$.

3. Применение операций математической морфологии: операция замыкания – для удаления разрывов на разделяющих парковочные места полосах, размыкания – фильтрация мелких и несвязанных элементов.

Замыкание выполняется на основе правила:

$$A \bullet B = (A + B) - B,$$

где $A + B$ – операция дилатации; A – обрабатываемое изображение; B – структурирующий элемент типа «квадрат» размером 7×7 ; $A - B$ – операция эрозии.

Размыкание выполняется как

$$A \circ B = (A - B) + B,$$

где B – структурирующий элемент типа «квадрат» размером 5×5 .

4. Горизонтальное дифференцирование изображений. Изображению $I_{x,y}$ ($x = \overline{0, w-1}$, $y = \overline{0, h-1}$, где w – ширина изображения, h – высота изображения) ставится в соответствие конечное множество векторов $\{V_i(x, y, l)\}_{i=0, nov-1}$, где каждый вектор $V_i(x, y, l)$, из nov общего числа векторов, задает неразрывную горизонтальную последовательность пикселей, с началом в координате (x, y) и длиной l , а именно $(x, y) = (x, y+1) = \dots = (x, y+l-1) = Color.Black$. Множество векторов горизонтальных линий $\{V_i\}_{i=0, nov-1}$, определяется последовательностью следующих действий:

```

1   nov = 0, l = 0
2   for y = 0 : h - 1
3     for x = 0 : w - 1
4       if  $I_{x,y} \equiv Color.Black$ 
5          $V(x, y, l) = V(0, 0, 0)$ 
6         while  $I_{x,y} \equiv Color.Black$ 
7            $V : x = V : x + 1$ 
8            $V : l = V : l + 1$ 
9            $V_{nov}(x, y, l) = V(x, y, l)$ 
10          nov = nov + 1, l = 0

```

5. Построение вертикальных линий $\{L_j(\overline{V})\}_{j=0, nol-1}$ на основе элементов множества векторов $\{V_i\}$,

где nol – количество детектированных линий, осуществляется следующим образом:

```

1   j = 0, nov_j = 0 (nov_j – количество векторов в линии)
2   T = V_j
3   while nov_j ≠ 0
4     nov_j = 0
5     L : V_{nov_j} = T
6     nov_j = nov_j + 1
7     for i = 0 : nov - 1
8       T = V_i
9       min = min(L : V_0 : l, L : V_{nov_j-1} : l, T : l)
10      if L : V_{nov_j-1} : y - T : y ≡ 1 and min > |L : V_0 : x - T : x|
and min > |L : V_0 : x + L : V_0 : l - T : x + T : l|
11      L : V_{nov_j} = T
12      nov_j = nov_j + 1

```

```

13    remove T from {Vi}
14    i = i - 1
15    if novj ≡ 1
16        remove T from {Vi}
17    else
18        Lj = L
19        j = j + 1

```

В вертикальной линии L_j , как указано в условии 10, образующие её вектора должны находиться один под другими и не превышать соседние по длине.

6. Описание линий начальными координатами (x, y) , шириной w и высотой h : $L_j(x, y, w, h)$. Для этого в область, образуемую последовательностью векторов, вписывается прямоугольник, с максимально возможной шириной и высотой. Результатом данного шага является детектирование вертикальных полос, разделяющих парковочные места.

7. Фильтрация линий по средней высоте и ширине. При вычислении средних значений не используются линии с высотой или шириной, равной единице.

8. Определение координат парковочных мест. Линии сортируются по начальным координатам: слева на право, сверху вниз. Затем они объединяются в парковочные места, образуя множество парковочных мест $\{S_k(L_l, L_r)\}_{k=0, nos-1}$, где L_l, L_r – левая и правая разделяющие полосы соответственно, nos – количество парковочных мест. Формирование множества парковочных мест $\{S_k\}$ происходит следующим образом:

```

1    i = 0, nos = 0
2    S = S(L0, L1)
3    i = i + 1 + 1
4    for i = i : nol - 1
5        if max(S : Ll : w, S : Lr : w, Li : w) > |(S : Lr : x - S : Ll : x) - (Li : x - S : Lr : x)|
6            Snos = S
7            nos = nos + 1
8            S = S(Snos-1 : Lr, Li)
9        else if i < nol - 1
10           S = S(Li, Li+1)
11           i = i + 1

```

9. Описание парковочного места S_k начальными координатами (x, y) , шириной w , высотой h : $\{S_k(x, y, w, h)\}_{k=0, nos-1}$, при этом

$$x = L_l : x + L_l : w;$$

$$y = \min(L_l : y, L_r : y);$$

$$w = L_r : x - x;$$

$$h = \max(L_l : y + L_l : h, L_r : y + L_r : h) - y.$$

2. Программная реализация и результаты исследования. Для проведения экспериментов и оценки результатов исследований рассмотренный подход реализован на языке Java с использованием библиотеки компьютерного зрения OpenCV 3.0.0. При этом реализованы следующие классы.

Element предназначен для хранения информации о горизонтальных векторах $V_i(x, y, l)$. В качестве свойств класса используются переменные x, y , и l , которые описывают последовательность пикселей длиной l с началом в координате (x, y) . Конструктор *Element* производит инициализацию полей класса.

ElementsLine применяется для описания вертикальной полосы $L_j(\bar{V})$ набором векторов $\{V_i\}$. Класс наследует `ArrayList<Element>` и представляет собой список элементов *Element*. *ElementsLine* содержит координаты (sx, sy) начала, координаты (ex, ey) окончания линии и параметры размеров (w, h)

линии. Метод $calculate\$x\$y\$w\h осуществляет определение размеров линии по имеющимся элементам текущего списка и переводит класс в вид $L_j(x, y, w, h)$, где (x, y) – координаты верхнего левого угла.

$Lines$ представляет собой список элементов $ElementsLine$, наследуя $ArrayList<ElementsLine>$, и содержит метод $defineElementsLine$, который определяет параметры линий по координатам ее начала и размерам для всех содержащихся в списке элементов.

$ParkingSpace$ предназначен для хранения информации о парковочном месте, которое описывается левой и правой разделяющими линиями $S_k(L_l, L_r)$, параметрами начальной точки и размером парковочного места $S_k(x, y, w, h)$. Полями класса являются левая и правая разделяющие парковочные линии ($left, right$), координаты места парковки (sx, sy, ex, ey) , размеры парковочного места (w, h) . Класс содержит конструктор $ParkingSpace$, осуществляющий инициализацию его свойств.

$ParkingSpaceGroup$ представляет собой список элементов $ParkingSpace$, наследуя $ArrayList<ParkingSpace>$, и хранит информацию о местах парковки в парковочном ряду. Полями класса являются координаты и размеры парковочного ряда. Конструктор $ParkingSpaceGroup$ осуществляет инициализацию свойств класса. Метод add является переопределенной функцией интерфейса $List$, предназначен для определения изменений свойств класса при добавлении нового региона интереса.

$PakingLot$ является вершиной иерархии описания парковки, наследует $ArrayList<ParkingSpaceGroup>$. Данный класс содержит методы $read$ и $write$. Метод $read$ считывает данные, описывающие регионы интереса и их статусы на изображении парковки, и возвращает элемент класса $PakingLot$. Метод $write$ предназначен для записи элемента $PakingLot$ в файл.

$PakingSpaceDetector$ используется для локализации регионов интереса парковочных мест согласно алгоритму, описанному в разделе 1. Для выполнения шагов 1–3, 4–6 и 7 алгоритма реализованы функции $processImage$, $detectVecticalLines$ и $detectParkingSpaces$ соответственно. Метод $detect$ последовательно вызывает указанные функции. Метод предназначен $definePakingLot$ для объединения парковочных мест в ряды.

Для осуществления чтения и записи данных парковки в/из файлов формата *.xml реализованы классы $XMLParkingLotReader$ и $XMLParkingLotWriter$.

На рисунке 1 представлен пример работы основных шагов предложенного алгоритма. В результате последовательного применения преобразования перспективы изображения, бинаризации методом Отсу и операций математической морфологии для исходного изображения (рис. 2, а) получаем изображение парковки, показанное на рисунке 1, а.

Рисунок 1, б представляет собой увеличенный фрагмент изображения, выделенный красным цветом на рисунке 1, а.

Результат горизонтального дифференцирования изображения (рис. 1, б) показан на рисунке 1, в.

На основе горизонтального дифференцирования построены вертикальные линии (рис. 1, г), которые объединяются в парковочные места (рис. 1, д).

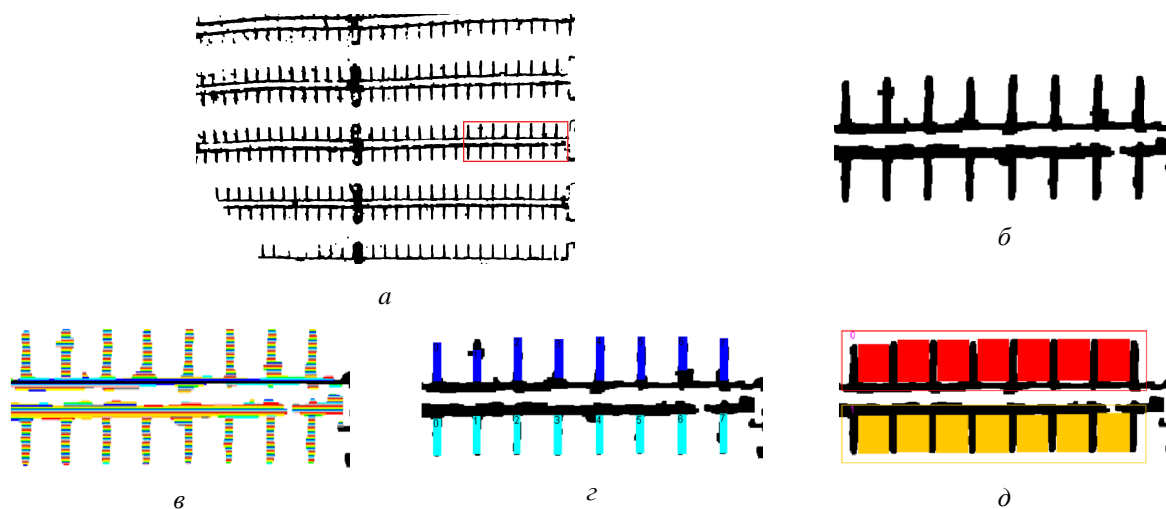


Рисунок 1 – Пример визуализации основных шагов алгоритма сегментации парковочных мест:
 а – результат применения шагов 1–3 алгоритма; б – увеличенный фрагмент изображения из рисунка 1 а;
 в – горизонтальное дифференцирование изображения; г – результат построения вертикальных линий;
 д – сегментированные парковочные места

Анализ результатов исследований свидетельствует о том, что предложенный алгоритм позволяет достаточно эффективно сегментировать парковочные места на изображениях автостоянок. Ошибки в работе алгоритма возможны в случае существенного отличия размеров парковочных мест или их перекрытия сторонними объектами.

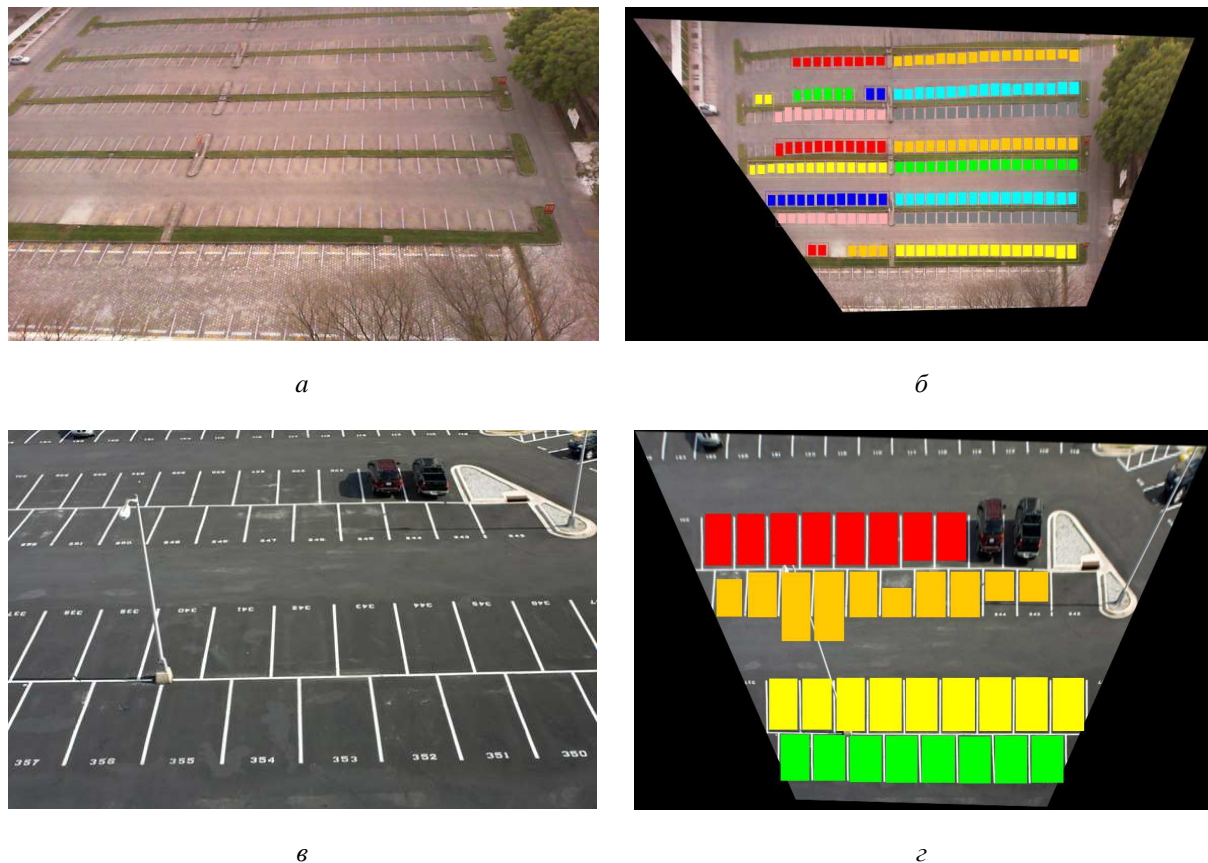


Рисунок 2 – Пример сегментации мест парковки:
***a* и *в* – исходные изображения парковок, *б* и *г* – результат сегментации парковочных мест**

Заключение. Синтезирован алгоритм сегментации парковочных мест, который требует выполнения следующих основных шагов:

- преобразования перспективы исходного изображения;
- адаптивной бинаризации методом Отсу;
- применения операций математической морфологии;
- горизонтального дифференцирования изображения;
- построения вертикальных линий;
- фильтрации линий по толщине;
- определения координат парковочных мест.

Для проведения экспериментов и оценки результатов исследований рассмотренный подход реализован на языке *Java* с использованием библиотеки компьютерного зрения *OpenCV 3.0.0*. Анализ результатов исследований свидетельствует о том, что предложенный алгоритм позволяет достаточно эффективно сегментировать и локализовать парковочные места на изображениях автомобильных парковок.

ЛИТЕРАТУРА

1. Алпатов, Б. А. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений в управлении / Б. А. Алпатов [и др.]. – М. : Радиотехника, 2008. – 176 с.
2. Богуш, Р. П. Система обработки информации для интеллектуального видеонаблюдения / Р. П. Богуш [и др.] // Вест. Полоц. гос. ун-та. Сер. С, Фундамент. науки. – 2010. – № 3. – С. 23–29.

3. Baroffio, L. A visual sensor network for parking lot occupancy detection in smart cities / L. Baroffio [et al.] // Internet of things, IEEE 2nd world forum. – 2015. – P. 745–750.
4. Wang, X. Parking lot analysis and visualization from aerial image / X. Wang, A. Hanson // Proc. 4th IEEE Workshop on Applications of Computer Vision. – 1998. – P. 36–41.
5. Bong, D. B. L. Integrated approach in the design of car park occupancy information system. / D. B. L. Bong, K. C. Ting, K. C. Lai // IAENG International Journal of Computer Science. – 2008. – № 35. – P. 1–8.
6. Seo, Y. W. Utilizing prior information to enhance self-supervised aerial image analysis for extracting parking lot structures / Y. W. Seo, C. Urmson // 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. – 2009. – P. 339–344.
7. Otsu, N. A threshold selection method from gray-level histogram / N. Otsu // IEEE Transactions on Systems, Man and Cybernetics. – 1979. – № 9. – P. 62–66.

Поступила 16.09.2016

PARKING SPACES SEGMENTATION IN IMAGES OF CAR PARKING

P. YARASHEVICH, R. BOGUSH, D. HLUKHAU

The algorithm includes: perspective transformation of the original image, he adaptive binarization Otsu's method, application of mathematical morphology operations, horizontal image differentiation, construction of vertical lines, filtering, determination of coordinates of parking spaces. For the experiments performed software implementation of the algorithm, described implemented classes. The results of the experiments are proposed.

Keywords: machine vision, image analysis, segmentation of parking spots.