

Нижегородский государственный университет им. Н.И. Лобачевского

Факультет вычислительной математики и кибернетики

Учебный курс
«Разработка мультимедийных приложений
с использованием библиотек OpenCV и IPP»

Лекция
Детекторы и дескрипторы ключевых точек.
Алгоритмы классификации изображений.
Задача детектирования объектов на
изображениях и методы её решения

Половинкин П.Н.

При поддержке компании Intel

Нижний Новгород

2013

Содержание

1.	АЛГОРИТМЫ ДЕТЕКТИРОВАНИЯ КЛЮЧЕВЫХ ТОЧЕК.....	3
1.1.	ДЕТЕКТОР МОРАВЕЦА.....	3
1.2.	ДЕТЕКТОР ХАРРИСА И СТЕФАНА.....	4
1.3.	ДЕТЕКТОР MSER's.....	5
1.4.	ДЕТЕКТОР FAST.....	6
2.	ОБЗОР ДЕСКРИПТОРОВ КЛЮЧЕВЫХ ТОЧЕК	7
2.1.	ДЕСКРИПТОР SIFT.....	8
2.2.	ДЕСКРИПТОР PCA-SIFT.....	9
2.3.	ДЕСКРИПТОР SURF.....	9
2.4.	ДЕСКРИПТОР GLOH.....	10
2.5.	ДЕСКРИПТОР DAISY.....	11
2.6.	ДЕСКРИПТОР BRIEF.....	12
3.	ЗАДАЧА КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ И МЕТОДЫ ЕЁ РЕШЕНИЯ.....	13
3.1.	АЛГОРИТМЫ КЛАССА BAG-OF-WORDS.....	13
3.2.	СОПОСТАВЛЕНИЕ ПРОСТРАНСТВЕННЫХ ПИРАМИД.....	14
3.3.	МОДЕЛИ ОБЪЕКТОВ, ОСНОВАННЫЕ НА ЧАСТЯХ	16
4.	ЗАДАЧА ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ И МЕТОДЫ ЕЁ РЕШЕНИЯ.....	18
4.1.	ПОСТАНОВКА ЗАДАЧИ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ	18
4.2.	МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ.....	18
4.2.1.	Методы, основанные на извлечении признаков.....	19
4.2.2.	Методы поиска по шаблону	20
5.	ЛИТЕРАТУРА	22
5.1.	ОСНОВНАЯ ЛИТЕРАТУРА.....	22
5.2.	РЕСУРСЫ СЕТИ ИНТЕРНЕТ	37

1. Алгоритмы детектирования ключевых точек

1.1. Детектор Моравца

Одним из наиболее распространенных типов особых точек являются углы на изображении [104], т.к. в отличие от ребер углы на паре изображений можно однозначно сопоставить. Расположение углов можно определить, используя локальные детекторы. Входом локальных детекторов является черно-белое изображение. На выходе формируется матрица с элементами, значения которых определяют степень правдоподобности нахождения угла в соответствующих пикселях изображения. Далее выполняется отсечение пикселей со степенью правдоподобности, меньшей некоторого порога. Для оставшихся точек принимается, что они являются особыми.

Детектор Моравца (Moravec) [148] является самым простым детектором углов. Автор данного детектора предлагает измерять изменение интенсивности пикселя (x, y) посредством смещения небольшого квадратного окна с центром в (x, y) на один пиксель в каждом из восьми принципиальных направлений (2 горизонтальных, 2 вертикальных и 4 диагональных). Размер окна обычно выбирается равным 3×3 , 5×5 или 9×9 пикселей. Детектор работает в несколько шагов:

- Для каждого направления смещения $(u, v) \in \{(1, 0), (1, 1), (0, 1), (-1, 1), (-1, 0), (-1, -1), (0, -1), (1, -1)\}$ вычисляется изменение интенсивности:

$$V_{u,v}(x, y) = \sum_{\substack{\forall a, b, \text{расположенных} \\ \text{в окне}}} (I(x + u + a, y + v + b) - I(x + a, y + b))^2$$

где $I(x, y)$ – интенсивность пикселя с координатами (x, y) в исходном изображении.

- Строится карта вероятности нахождения углов в каждом пикселе изображения посредством вычисления оценочной функции $C(x, y) = \min\{V_{u,v}(x, y)\}$. По существу определяется направление, которому соответствует наименьшее изменение интенсивности, т.к. угол должен иметь смежные ребра.
- Отсекаются пиксели, в которых значения оценочной функции ниже некоторого порогового значения.
- Удаляются повторяющиеся углы с помощью процедуры NMS (non-maximal suppression) [85]. Все полученные ненулевые элементы карты соответствуют углам на изображении.

Основными недостатками рассматриваемого детектора являются отсутствие инвариантности к преобразованию типа «поворот» и возникновение ошибок детектирования при наличии большого количества

диагональных ребер. Очевидно, что детектор Моравеца обладает свойством анизотропии в 8 принципиальных направлениях смещения окна.

1.2. Детектор Харриса и Стефана

Детектор Харриса (Harris) [105, 104, 109, 149] строится на основании детектора Моравеца и является его улучшением, т.к. для него характерна анизотропия по всем направлениям. Харрис и Стефан (Stephen) вводят в рассмотрение производные по некоторым принципиальным направлениям, раскладывают функцию интенсивности в ряд Тейлора:

$$\begin{aligned} I(x + u + a, y + v + b) &\approx I(x + a, y + b) + u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} \\ &= I(x + a, y + b) + \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}. \end{aligned}$$

Как следствие, изменение интенсивности $V_{u,v}(x, y)$ в каждом пикселе можно рассматривать как функцию следующего вида:

$$\begin{aligned} V_{u,v}(x, y) &= \sum_{\substack{\forall a, b, \text{расположенных} \\ \text{в окне}}} \left(\begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right)^2 \\ &= \sum_{\substack{\forall a, b, \text{расположенных} \\ \text{в окне}}} \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\ &= \begin{bmatrix} u & v \end{bmatrix} \left(\sum_{\substack{\forall a, b, \text{расположенных} \\ \text{в окне}}} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \\ &= \begin{bmatrix} u & v \end{bmatrix} A_{u,v}(x, y) \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

В матрице Харриса $A_{u,v}(x, y)$ (автокорреляционная матрица), как правило, выбирают взвешенную свертку производных с весовыми коэффициентами Гауссова окна (окно размером 3×3 пикселя с коэффициентами 0.04, 0.12, 0.04, 0.12, 0.36, 0.12, 0.04, 0.12, 0.04 – представление по строкам). Отметим, что матрица Харриса симметричная и положительно полуопределенная. Вычисляя собственные значения полученной матрицы [105], точки изображения можно классифицировать на ребра и углы:

- если оба собственных числа автокорреляционной матрицы достаточно большие, т.е. небольшой сдвиг окна приводит к значительным изменениям интенсивности, то пиксель классифицируется как угол;

- если одно собственное число значительно больше другого, то это означает, что окно было сдвинуто перпендикулярно выступу, поэтому пиксель принадлежит ребру;
- если собственные значения близки к нулю, тогда текущий пиксель не содержит ни углов, ни ребер.

Детектор Харриса по сравнению с ранее рассмотренным детектором требует *большого* количества вычислений за счет необходимости построения сверток с Гауссовым ядром. При этом он достаточно восприимчив к шумам. Подавить шумы позволяет увеличение размера Гауссова окна, но это приводит к значительным вычислительным расходам, поэтому необходимо находить компромисс между качеством работы алгоритма и количеством выполняемых операций. Детектор Харриса обладает свойством анизотропии вдоль горизонтального и вертикального направлений, т.к. автокорреляционная матрица содержит первые производные только вдоль указанных направлений. По сравнению со своим предшественником данный детектор инвариантен относительно поворота, количество ошибок детектирования углов не велико за счет введения свертки с Гауссовыми весовыми коэффициентами. Результаты детектирования значительно меняются при масштабировании изображения. Впоследствии возникают модификации детектора Харриса, которые учитывают вторые производные функции интенсивности (например, детектор Харриса-Лапласа (Harris-Laplace) [78]).

1.3. Детектор MSER's

При разработке детектора MSER (Maximally Stable Extremal Regions, Matas и др., 2002) решается проблема инвариантности особых точек при масштабировании изображения. Детектор [19, 104, 75, 109] выделяет множество различных регионов с экстремальными свойствами функции интенсивности внутри региона и на его внешней границе.

Рассмотрим идею алгоритма для случая черно-белого изображения. Представим все возможные отсечения изображения. В результате получим набор бинарных изображений при разных значениях порога $\tau \in \{0, 1, \dots, 255\}$ (пиксель, интенсивность которого меньше порога считаем черным, в противном случае, белым). Таким образом, строится пирамида, у которой на начальном уровне, соответствующем минимальному значению интенсивности, находится белое изображение, а на последнем уровне, отвечающем максимальному значению интенсивности, – черное. Если в некоторый момент происходит движение, то на белом изображении появляются черные пятна, соответствующие локальным минимумам интенсивности. С увеличением порога пятна начинают разрастаться и сливаться, в конечном итоге образуя единое черное изображение. Такая пирамида позволяет построить множество связанных компонент, соответствующих белым областям, – регионов с максимальным значением

интенсивности. Если инвертировать бинарные изображения в пирамиде, то получим набор регионов с минимальным значением интенсивности. Схема алгоритма состоит из нескольких этапов:

- Отсортируем множество всех пикселей изображения в порядке возрастания/убывания интенсивности. Отметим, что такая сортировка возможна за время, пропорциональное количеству пикселей.
- Построим пирамиду связных компонент. Для каждого пикселя отсортированного множества выполним последовательность действий:
- обновление списка точек, входящих в состав компоненты;
- обновление областей следующих компонент, в результате чего пиксели предыдущего уровня будут подмножеством пикселей следующего уровня.
- Выполним для всех компонент поиск локальных минимумов (находим пиксели, которые присутствуют в данной компоненте, но не входят в состав предыдущих). Набор локальных минимумов уровня соответствует экстремальному региону на изображении.

1.4. Детектор FAST

Описанные ранее детекторы определяют особые точки на изображении, в частности, углов, применяя некоторую модель или алгоритм напрямую к пикселям исходного изображения. Альтернативный подход состоит в том, чтобы использовать алгоритмы машинного обучения для тренировки классификатора точек на некотором множестве изображений. FAST-детектор (Features from Accelerated Test) [96, 109] строит деревья решений для классификации пикселей.

Для каждого пикселя p изображения рассматривается окружность с центром в этой точке, которая вписана в квадрат со стороной 7 пикселей (Рисунок 1). Окружность проходит через 16 пикселей окрестности.

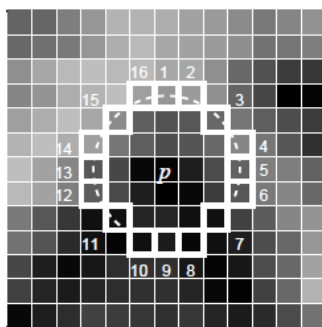


Рисунок 1 – Рабочая окрестность пикселя при использовании FAST детектора

Каждый окрестный пиксель $x \in \{1, 2, \dots, 16\}$ относительно центрального ($p \rightarrow x$) может находиться в одном из трех состояний:

$$S_{p \rightarrow x} = \begin{cases} d, & I_x \leq I_p - t \text{ (темнее)} \\ s, & I_p - t < I_x < I_p + t \text{ (похожий)} \\ b, & I_p + t \leq I_x \text{ (светлее)} \end{cases}$$

Выбирая x и вычисляя $S_{p \rightarrow x}$ для каждого $p \in P$ – множества всех пикселей тренировочного набора изображений – разделяем множество P на три подмножества P_d, P_s, P_b – множества точек, которые темнее, схожи и светлее x соответственно. Далее выполняется построение дерева решений согласно алгоритму, описанному в [93]. На каждом уровне дерева решений множество, соответствующее узлу дерева, разбивается на подмножества посредством выбора наиболее информативной точки (пикселя с наибольшей энтропией). Построенное дерево решений в результате используется для определения углов на тестовых изображениях.

2. Обзор дескрипторов ключевых точек

Результатом работы детекторов является множество особых точек, для которых необходимо построить математическое описание. В настоящем разделе делается обзор некоторых существующих дескрипторов.

Входными данными дескриптора является изображение и набор особых точек, выделенных на заданном изображении. Выходом дескриптора является множество векторов признаков для исходного набора особых точек. Необходимо отметить, что какие-то дескрипторы решают одновременно две задачи – поиск особых точек и построение описателей этих точек.

Признаки (описатели) строятся на основании информации об интенсивности, цвете и текстуре особой точки. Но особые точки могут представляться углами, ребрами или даже контуром объекта, поэтому, как правило, вычисления выполняются для некоторой окрестности. В идеале хорошие признаки должны обладать рядом свойств:

- Повторяемость. На изображениях одного и того же объекта или сцены, сделанных с разных точек зрения и при разных условиях освещенности, большинство признаков должно быть продетектировано.
- Локальность. Признаки должны быть максимально локальными, чтобы снизить вероятность перекрытий.
- Репрезентативность. Количество признаков должно быть достаточным, чтобы разумное число признаков детектировалось даже на небольшом изображении объекта.

- Точность. Признаки должны быть точно продетектированными по отношению к масштабу и форме объекта.
- Эффективность. Для приложений реального времени критично, чтобы процедура вычисления признаков не требовала значительных вычислительных затрат.

2.1. Дескриптор SIFT

Для формирования дескриптора SIFT (Scale Invariant Feature Transform) [105, 73, 109] сначала вычисляются значения магнитуды и ориентации градиента в каждом пикселе, принадлежащем окрестности особой точки размером 16×16 пикселей. Магнитуды градиентов при этом учитываются с весами, пропорциональными значению функции плотности нормального распределения с математическим ожиданием в рассматриваемой особой точке и стандартным отклонением, равным половине ширины окрестности (веса Гауссова распределения используются для того, чтобы уменьшить влияние на итоговый дескриптор градиентов, вычисленных в пикселях, находящихся дальше от особой точки).

В каждом квадрате размером 4×4 пикселя вычисляется гистограмма ориентированных градиентов путем добавления взвешенного значения магнитуды градиента к одному из 8 бинов гистограммы. Чтобы уменьшить различные «граничные» эффекты, связанные с отнесением похожих градиентов к разным квадратам (что может возникнуть вследствие небольшого сдвига расположения особой точки) используется билинейная интерполяция: значение магнитуды каждого градиента добавляется не только в гистограмму, соответствующую квадрату, к которому данный пиксель относится, но и к гистограммам, соответствующим соседним квадратам. При этом значение магнитуды добавляется с весом, пропорциональным расстоянию от пикселя, в котором вычислен данный градиент, до центра соответствующего квадрата. Все вычисленные гистограммы объединяются в один вектор, размером, равным $128 = 8(\text{число бинов}) \times 4 \times 4(\text{число квадратов})$.

Полученный дескриптор преобразуется, чтобы уменьшить возможные эффекты от изменения освещенности. Изменение контраста изображения (значение интенсивности каждого пикселя умножается на некоторую константу) приводит к такому же изменению в значениях магнитуд градиентов. Поэтому очевидно, что данный эффект может быть нивелирован путем нормализации дескриптора таким образом, чтобы его длина стала равна единице. Изменения яркости изображения (к значению интенсивности каждого пикселя прибавляется некоторая константа) не влияют на значения магнитуд градиентов. Таким образом, SIFT-дескриптор является инвариантным по отношению к аффинным изменениям освещенности. Однако могут возникать и нелинейные изменения в освещенности вследствие, например, различной ориентации

источника света по отношению к поверхностям трехмерного объекта. Данные эффекты могут вызвать большое изменение в отношении магнитуд некоторых градиентов (при этом оказывают незначительное влияние на ориентацию вектора градиента). Чтобы избежать этого, используют отсечение по некоторому порогу (по результатам экспериментов показано, что оптимальным является значение 0.2), которое применяют к компонентам нормализованного дескриптора. После применения порога дескриптор вновь нормализуется. Таким образом, уменьшается значение больших магнитуд градиентов и увеличивается значение распределения ориентаций данных градиентов в окрестности особой точки.

2.2. Дескриптор PCA-SIFT

Дескриптор PCA-SIFT [105, 64] по существу является модификацией SIFT. На начальном этапе аналогично вычисляются значения магнитуды и ориентации градиента. Только для каждой особой точки рассматривается окрестность размером 41×41 пиксель с центром в точке, которая является особой. По факту строится карта градиентов вдоль вертикального и горизонтального направлений. Как следствие, получается вектор, содержащий $2 \times 39 \times 39 = 3042$ элементов. Далее выполняется построение SIFT дескриптора согласно схеме, описанной в предыдущем разделе. Для результирующего набора SIFT дескрипторов осуществляется снижение размерности векторов до 32 элементов посредством анализа главных компонент (Principal Component Analysis, PCA).

2.3. Дескриптор SURF

Дескриптор SURF (Speeded up Robust Features) [14, 109] относится к числу тех дескрипторов, которые одновременно выполняют поиск особых точек и строят их описание, инвариантное к изменению масштаба и вращения. Кроме того, сам поиск ключевых точек обладает инвариантностью в том смысле, что повернутый объект сцены имеет тот же набор особых точек, что и образец.

Определение особых точек на изображении выполняется на основании матрицы Гессе (FAST-Hessian detector, [109]). Использование Гессиана обеспечивает инвариантность относительно преобразования типа «поворот», но не инвариантность относительно изменения масштаба. Поэтому SURF применяет фильтры разного масштаба для вычисления Гессиана. Предположим, что исходное изображение задается матрицей интенсивностей I , текущий рассматриваемый пиксель обозначим через $X = (x, y)$, а σ – масштаб фильтра. Тогда матрица Гессе имеет вид:

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix},$$

где $L_{xx}(X, \sigma)$, $L_{xy}(X, \sigma)$, $L_{yy}(X, \sigma)$ – свертки аппроксимации второй производной Гауссова ядра с изображением I . Детерминант матрицы Гессе достигает экстремума в точках максимального изменения градиента яркости. Поэтому SURF пробегается фильтром с Гауссовым ядром по всему изображению и находит точки, в которых достигается максимальное значение детерминанта матрицы Гессе. Отметим, что такой проход выделяет как темные пятна на белом фоне, так и светлые пятна на темном фоне.

Далее для каждой найденной особой точки вычисляется ориентация – преобладающее направление перепада яркости. Понятие ориентации близко к понятию направления градиента, но для определения ориентации особой точки применяется фильтр Хаара [114].

На основании имеющейся информации выполняется построение дескрипторов для каждой особой точки:

- Вокруг точки строится квадратная окрестность размером $20s$, где s – масштаб, на котором получено максимальное значение детерминанта матрицы Гессе.
- Полученная квадратная область разбивается на блоки, в результате область будет разбита на 4×4 региона.
- Для каждого блока вычисляются более простые признаки. Как следствие, получается вектор, содержащий 4 компоненты: 2 – это суммарный градиент по квадранту, 2 – сумма модулей точечных градиентов.
- Дескриптор формируется в результате склеивания взвешенных описаний градиента для 16 квадрантов вокруг особой точки. Элементы дескриптора взвешиваются на коэффициенты Гауссова ядра. Веса необходимы для большей устойчивости к шумам в удаленных точках.
- Дополнительно к дескриптору заносится след матрицы Гессе. Эти компоненты необходимы, чтобы различать темные и светлые пятна. Для светлых точек на темном фоне след отрицателен, для темных точек на светлом фоне – положителен.

Отметим, что SURF используется для поиска объектов. Тем не менее, дескриптор никак не использует информацию об объектах. SURF рассматривает изображение как единое целое и выделяет особенности всего изображения, поэтому он плохо работает с объектами простой формы.

2.4. Дескриптор GLOH

Дескриптор GLOH (Gradient location-orientation histogram) [77] является модификацией SIFT-дескриптора, который построен с целью повышения

надежности. По факту вычисляется SIFT дескриптор, но используется полярная сетка разбиения окрестности на бины (Рисунок 2): 3 радиальных блока с радиусами 6, 11 и 15 пикселей и 8 секторов. В результате получается вектор, содержащий 272 компоненты, который проецируется в пространство размерности 128 посредством использования анализа главных компонент (PCA).

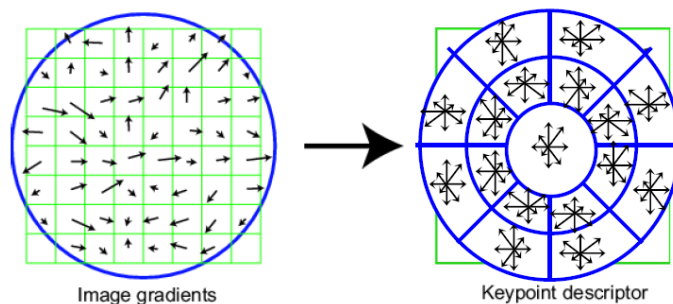


Рисунок 2 – Полярная сетка разбиения на бины [105]

2.5. Дескриптор DAISY

Детектор DAISY [107] изначально вводится для решения задачи сопоставления изображений (matching) в случае значительных внешних изменений, т.е. данный дескриптор в отличие от ранее рассмотренных работает на плотном множестве пикселей всего изображения. При этом авторы DAISY в работе [107] показали, что дескриптор работает в 66 раз быстрее, чем SIFT, запущенный на плотном множестве пикселей. В DAISY использованы идеи построения SIFT и GLOH дескрипторов. Аналогично GLOH выбирается круговая окрестность особой точки, при этом бины представляются не частичными секторами, а окружностями (Рисунок 3).

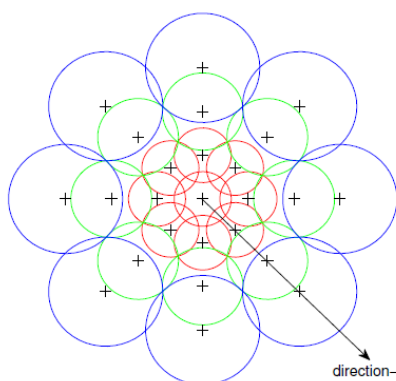


Рисунок 3 – Сетка построения бинов для центрального пикселя [107]

Для каждого такого бина выполняется та же последовательность действий, что и в алгоритме SIFT, но взвешенная сумма магнитуд градиентов заменяется сверткой исходного изображения с производными Гауссова фильтра, взятыми по 8 направлениям. Авторы [107] показали, что построенный дескриптор обладает инвариантностью, как и SIFT, и GLOH, при этом для решения задачи сопоставления (matching) в случае, когда все пиксели считаются особыми, требует меньших вычислительных затрат.

2.6. Дескриптор BRIEF

Цель создания BRIEF-дескриптора (Binary Robust Independent Elementary Features) [22] состояла в том, чтобы обеспечить распознавание одинаковых участков изображения, которые были сняты с разных точек зрения. При этом ставилась задача максимально уменьшить количество выполняемых вычислений. Алгоритм распознавания сводится к построению случайного леса (randomize classification trees) или наивного Байесовского классификатора на некотором тренировочном множестве изображений и последующей классификации участков тестовых изображений. В упрощенном варианте может использоваться метод ближайшего соседа для поиска наиболее похожего патча в тренировочной выборке. Небольшое количество операций обеспечивается за счет представления вектора признаков в виде бинарной строки, а как следствие, использования в качестве меры сходства расстояния Хэмминга.

Схема построения векторов признаков достаточно простая. Изображение разбивается на патчи (отдельные перекрывающиеся участки). Допустим патч P имеет размеры $S \times S$ пикселей. Из патча выбирается некоторым образом множество пар пикселей $\{(X, Y), \forall X, Y \text{ в окрестности}\}$, для которых строится набор бинарных тестов:

$$\tau(P, X, Y) = \begin{cases} 1, & I(X) < I(Y) \\ 0, & \text{иначе} \end{cases},$$

где $I(X)$ – интенсивность пикселя X . Для каждого патча выбирается множество, содержащее n_d пар точек, которые однозначно определяют набор бинарных тестов. Далее на основании этих тестов строится бинарная строка:

$$f_{n_d}(P) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(P, X_i, Y_i)$$

Авторы [22] приводят результаты экспериментов (качества распознавания) при выборе пар точек согласно закону равномерного распределения в патче, а также нормального распределения с разными значениями математического ожидания и среднеквадратичного отклонения. Отметим, что при одинаковых условиях проведения экспериментов на некоторых

тестовых изображениях точность детектирования с помощью BRIEF почти в 1.5 раза выше, чем с использованием SURF-дескрипторов.

3. Задача классификации изображений и методы её решения

3.1. Алгоритмы класса bag-of-words

К одному из наиболее распространенных классов алгоритмов классификации изображений можно отнести так называемые bag-of-words (также известные как bag-of-features или bag-of-keypoints) методы, впервые предложенные в работе [27]. Фактически bag-of-words использует в качестве описания гистограмму вхождений отдельных шаблонов в изображение. Корни идеи восходят к задаче классификации текстов, для решения которой используют описания в виде гистограмм вхождений в документ слов из заранее составленного словаря.

Корни идеи восходят к задаче классификации текстов, для решения которой используют описания в виде гистограмм вхождений в документ слов из заранее составленного словаря.

Основные шаги алгоритмов класса bag-of-words могут быть описаны следующим образом:

1. Детектирование ключевых точек на изображении
2. Построение описания (дескрипторов) локальных окрестностей ключевых точек
3. Кластеризация дескрипторов ключевых точек, принадлежащих всем объектам обучающей выборки (это соответствует построению словаря, «словами» в котором являются центроиды построенных кластеров)
4. Построение описания каждого изображения в виде нормированной гистограммы встречаемости «слов» (для каждого кластера вычисляется количество отнесенных к нему ключевых точек, принадлежащих определенному изображению)
5. Построение классификатора, использующего вычисленное на шаге 4 признаковое описание изображения.

На дескрипторы, используемые алгоритмами данного класса, накладываются определенные ограничения: в частности, они должны быть инварианты к аффинным преобразованиям изображения, изменениям в условиях освещенности и окклюзиям. Словарь дескрипторов ключевых точек должен быть достаточно большим, чтобы отражать релевантные

изменения частей изображения, но в то же время не чрезмерным, чтобы сделать алгоритм устойчивым к шуму.

Одним из недостатков классического bag-of-words подхода является то, что он никак не учитывает пространственную информацию о распределении ключевых точек на изображении. Это приводит к тому, что описания объектов со схожими по дескрипторам наборами ключевых точек, находящимися в совершенно разных конфигурациях, совпадают. Для того чтобы учесть пространственную информацию, было предложено несколько подходов. Так, в работе [143] используются т.н. кореллограммы визуальных слов, позволяющие эффективно моделировать типовые пространственные корреляции между визуальными словами для определенных классов. В работе [144] рассматривается двухуровневая иерархическая модель, в которой объект представляется P частями, к каждой из которых относится N_p ключевых точек. Широкое развитие получил подход, в котором используется сопоставление пространственных пирамид признаков (де-факто, объект описывается не одной гистограммой, а объединением гистограмм, соответствующих отдельным частям изображения). Подробное изложение подхода приведено в разделе 3.2.

3.2. Сопоставление пространственных пирамид

Пусть X и Y – два множества векторов в d -мерном пространстве признаков (данные множества векторов соответствуют наборам дескрипторов ключевых точек). В работе [52] предложена схема сопоставления пирамид, которая находит приблизительное соответствие между двумя множествами такого типа. Данный алгоритм работает следующим образом: пространство признаков разбивается на последовательность вложенных друг в друга подобластей (ячеек) и вычисляется взвешенная сумма числа совпадений на всех уровнях разбиения. Для каждого фиксированного уровня две точки считаются соответствующими, если они принадлежат одной подобласти, совпадения на более детальных уровнях разбиения учитываются с большим весом, чем соответствия, найденные на более грубых уровнях разбиения.

Пусть имеется $(L + 1)$ уровней разбиения $0, \dots, L$, где l -ому уровню соответствует 2^l ячеек по каждой размерности (таким образом, общее число ячеек на l -ом уровне равно $D=2^{dl}$). Пусть H_X^l и H_Y^l – гистограммы, описывающие множества X и Y , в которых $H_X^l(i)$ и $H_Y^l(i)$ соответствует числу точек из X и Y , которые относятся к i -ой ячейке. Тогда число совпадений на l -ом уровне разбиения вычисляется с помощью функции пересечения гистограмм:

$$I(H_X^l, H_Y^l) = \sum_{i=1}^D \min(H_X^l(i), H_Y^l(i))$$

Необходимо отметить, что число совпадений, найденных на уровне l , также включает в себя число совпадений, найденных на уровне $l+1$, поэтому число новых совпадений, найденных на уровне l , вычисляется как $I^l - I^{l+1}$ для всех $l = 0, \dots, L-1$. Вес, соответствующий уровню l , устанавливается равным $\frac{1}{2^{L-l}}$, что обратно пропорционально ширине ячейки на этом уровне. С практической точки зрения кажется целесообразным установить меньший вклад в итоговую функцию числа совпадений совпадения, найденные на более грубых уровнях разбиений, т.к. они включают в себя явно непохожие друг на друга признаки.

Суммируя функции пересечения гистограмм на всех уровнях, получаем функцию ядра сопоставления пирамид:

$$\tau^L(X, Y) = I^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} (I^l - I^{l+1}) = \frac{1}{2^L} I^0 + \sum_{l=1}^L \frac{1}{2^{L-l+1}} I^l$$

Очевидно, что приведенная выше схема работает с неупорядоченным представлением изображения (функции, описанной выше, не важно расположение ключевых точек внутри изображения). Это позволяет точно сопоставить 2 набора признаков в многомерном пространстве, игнорируя при этом всю пространственную информации об их расположении на изображении. Авторами схемы сопоставления пространственных пирамид в работе [67] предложен альтернативный подход: выполнять сопоставление пирамид в двухмерном пространстве и использовать алгоритмы кластеризации в пространстве признаков. Все вектора признаков квантизируются к M дискретным типам (что соответствует отнесению вектора признаков к одному из M кластеров), и делается предположение, что только признаки одного типа могут соответствовать друг другу. Каждому значению $m = 1, \dots, M$ соответствует набор двухмерных векторов X_m и Y_m , представляющих координаты векторов признаков типа m , найденных на изображении. Итоговая функция ядра сопоставления пирамид является суммой отдельных ядер для каждого типа признаков:

$$K^L(X, Y) = \sum_{m=1}^M \tau^L(X_m, Y_m)$$

Так как функция ядра сопоставления пирамид является взвешенной суммой функций пересечения гистограмм, можно представить $K^L(X, Y)$ как единую функцию пересечения гистограмм для векторов признаков, полученных конкатенаций соответствующим образом взвешенных гистограмм, относящихся ко всем M типам признаков и ко всем уровням

пирамиды. Для L уровней и M типов признаков результирующий вектор будет иметь размерность $\frac{M}{3}(4^{L+1} - 1)$. Однако в силу разреженности гистограмм сложность вычисления ядра является линейной от числа признаков.

3.3. Модели объектов, основанные на частях

Одним из наиболее известных подходов к классификации объектов является использование моделей объектов, основанных на частях (part-based models). В отличие от bag-of-words методов с добавленной информацией о расположении ключевых точек алгоритмы данного класса явно учитывают *взаимное* расположение различных частей объекта. Впервые подход был представлен в работе [142] для задачи классификации лиц.

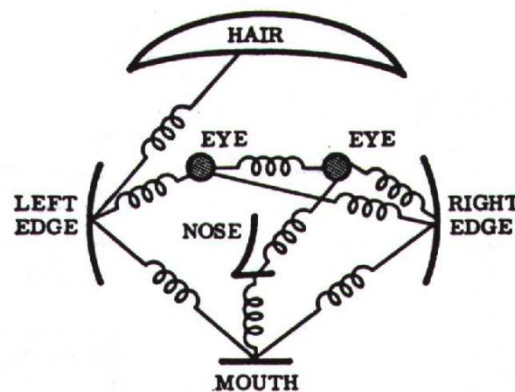


Рис. 1. Модель лица [142]

Основными элементами моделей объектов, основанных на частях, являются:

- представление (модель) отдельных частей объекта;
- методы обучения данного представления;
- описание связей между частями объекта.

Конфигурация, которой описывается взаимное расположение частей объекта, является одним из ключевых факторов, влияющих на качество и быстродействие для моделей данного класса. Если предположить, что объект содержит P частей, при этом для каждой части на изображении возможно N различных положений, то в данном случае имеется N^P различных конфигураций, что накладывает ограничение на практическую применимость методов в силу вычислительной сложности (при сопоставлении частей классифицируемого изображения с частями

обученной модели (т.н. *вывода* (inference)) приходится выполнить $O(N^P)$ операций. Ряд используемых моделей представлено на Рис. 2 [140]

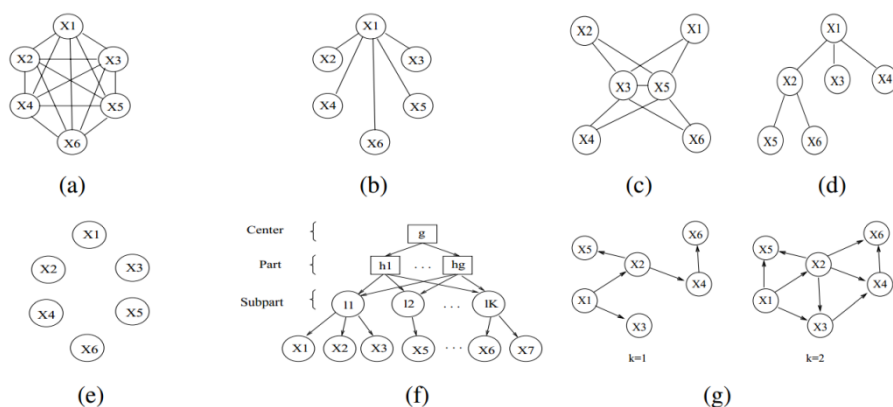


Рис. 2. Структура связей между частями объекта

- a) Constellation («созвездие») [135] (вычислительная сложность вывода $O(N^P)$)
- b) Star («звезда») [134] (вычислительная сложность вывода $O(N^2)$)
- c) k-fan [136] (вычислительная сложность $O(N^k)$, где k – число частей в клике графа, являющейся основой модифицированной структуры типа «звезда»)
- d) Tree («дерево») [137] (вычислительная сложность вывода $O(N^2(P-I)+NP)$ с при использовании основанного на динамическом программировании алгоритма Витерби [141])
- e) Bag-of-features [138]
- f) Hierarchy (иерархическая структура, описывающая зависимость между частями объекта и их объединениями (укрупненными частями)) [139]
- g) Sparse flexible models (направленный ациклический граф, описывающий отношение между частями объекта) [140]

Для представления отдельных частей объекта могут использоваться как дескрипторы ключевых точек (например, SIFT), так и другие признаки, например результаты PCA преобразования интенсивностей пикселей внутри патча, соответствующего части изображения, гистограммы ориентированных градиентов (HoG) [26] и т.д.

При обучении представления отдельных частей объекта используются как генеративные (например, Байесов классификатор), так и дискриминативные алгоритмы машинного обучения, которые отличаются лучшей точностью предсказания. В частности, в работе [39] был предложен подход, основанный на описании частей объекта в виде гистограмм ориентированных градиентов, для обучения представления

отдельных частей используется скрытый метод опорных векторов (Latent SVM).

4. Задача детектирования объектов на изображениях и методы её решения

4.1. Постановка задачи детектирования объектов

Задача детектирования решается человеческим зрением за считанные доли секунды. Для машины же такая задача является сложной. Сложность обусловлена многими причинами, среди которых можно выделить несколько основных:

1. Разнообразие форм и цветов представителей класса объектов. Например, в случае множества транспортных средств – это разнообразие моделей и цветов автомобилей.
2. Перекрывание детектируемых объектов. Естественная ситуация, когда с точки зрения камеры объекты видны только частично, например, обзревается только крыша и ветровое стекло автомобиля.
3. Разная степень освещенности объектов. В зависимости от времени суток одинаковые объекты могут выглядеть и восприниматься абсолютно по-разному.

Цель детектирования – определить наличие объекта на изображении и найти его положение в системе координат пикселей исходного изображения. Положение объекта в зависимости от выбора алгоритма детектирования может определяться координатами прямоугольника, охватывающего объект, либо контуром этого объекта, либо координатами точек, наиболее характерных для объекта.

Решение задачи детектирования объектов позволяет анализировать качественный состав сцены, представленной на изображении, а также получить информацию о взаимном расположении объектов.

4.2. Методы решения задачи детектирования объектов

Множество всех методов решения задачи детектирования можно разделить на три основные группы:

- Методы, которые для описания объекта используют признаки, наиболее характерные для объектов. В качестве признаков могут быть выбраны точечные особенности объекта, либо признаки, построенные для изображения, содержащего только объект.
- Методы поиска объектов, соответствующих шаблону – некоторому описанию объектов.

- Методы детектирования движения объектов – выделение движущихся объектов на основании нескольких изображений или кадров видео одной и той же сцены.

4.2.1. Методы, основанные на извлечении признаков

Один из возможных подходов к решению задачи детектирования состоит в том, чтобы использовать алгоритмы машинного обучения для построения моделей классов объектов и алгоритмы вывода для поиска объектов на изображении.

Построение модели состоит из двух этапов (Рисунок 4):

- Извлечение признаков, характерных для объектов класса, – построение характеристических векторов-признаков для ключевых точек объекта (углов, ребер [5] или контуров объектов [100]) или для всего объекта.
- Тренировка модели на полученных признаках для последующего распознавания объектов.

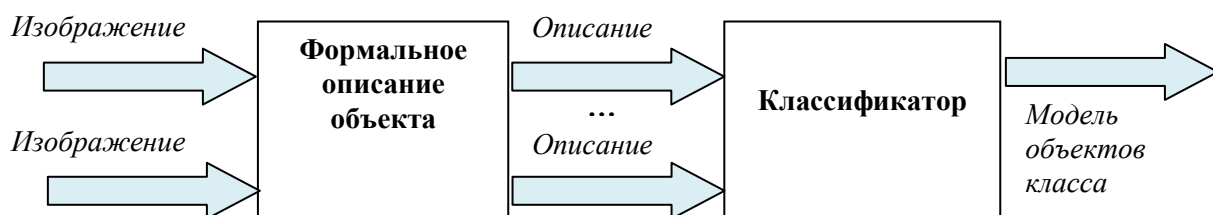


Рисунок 4 – Схема построения модели класса с использованием методов, основанных на извлечении характерных признаков

Техники данной группы описывают объект с использованием векторов-признаков. Вектора строятся на основании цветовой информации (гистограмма ориентированных градиентов (Histogram of Oriented Gradients или HOG) – один из наиболее популярных способов). Также может быть использована контекстная информация (context based) [108, 84], а в некоторых случаях – данные о геометрии и взаимном расположении частей объекта (part-based) [39]. Тем не менее, все эти методы строят некоторую математическую модель объекта на каждом изображении тренировочной выборки, содержащем объект. Формально признак x_i – это числовая характеристика. Для каждой ключевой точки алгоритмы данной группы строят вектор признаков (x_1, x_2, \dots, x_n) . Таким образом, объект описывается набором векторов признаков в характерных точках. В результате тренировки строится модель, содержащая «усредненные» вектора признаков.

Алгоритм вывода (поиска) по существу включает два этапа:

- Извлечение признаков объекта из тестового изображения. При извлечении признаков возникает две основные проблемы:
 - На изображении может быть много объектов одного класса, а необходимо найти всех представителей. Поэтому необходимо просматривать все части изображения, проходя «бегущим» окном (sliding window) от левого верхнего до правого нижнего угла. При этом размер окна определяется размером изображений тренировочной выборки.
 - Объекты на изображении могут иметь разный масштаб. Самое распространенное решение – масштабирование изображения.
- Поиск объектов на изображении (Рисунок 5). Входными данными алгоритма поиска являются формальное описание объекта – набор признаков, которые выделены из тестового изображения, – и модель класса объектов. На основании этой информации классификатор принимает решение о принадлежности объекта классу. Некоторые методы поиска также оценивают степень достоверности того, что объект принадлежит рассматриваемому классу.

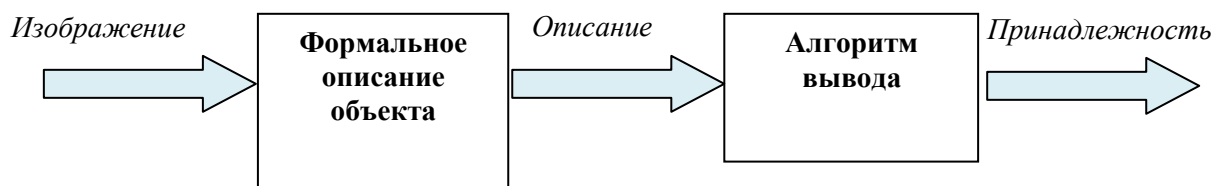


Рисунок 5 – Схема поиска объектов с использованием методов, основанных на извлечении характерных признаков

Качество рассматриваемых методов в основном зависит от того, насколько хорошо выбраны признаки, т.е. насколько хорошо эти признаки дифференцируют классы объектов. Существуют специализированные методы, основанные на извлечении признаков, для детектирования лиц [114, 113, 92], транспортных средств [4] и пешеходов [26, 115, 49].

4.2.2. Методы поиска по шаблону

Детектирование объектов на основании некоторого шаблона предполагает, что имеется изображение объекта с выделенными признаками – шаблон – и тестовое изображение, которое сопоставляется этому шаблону.

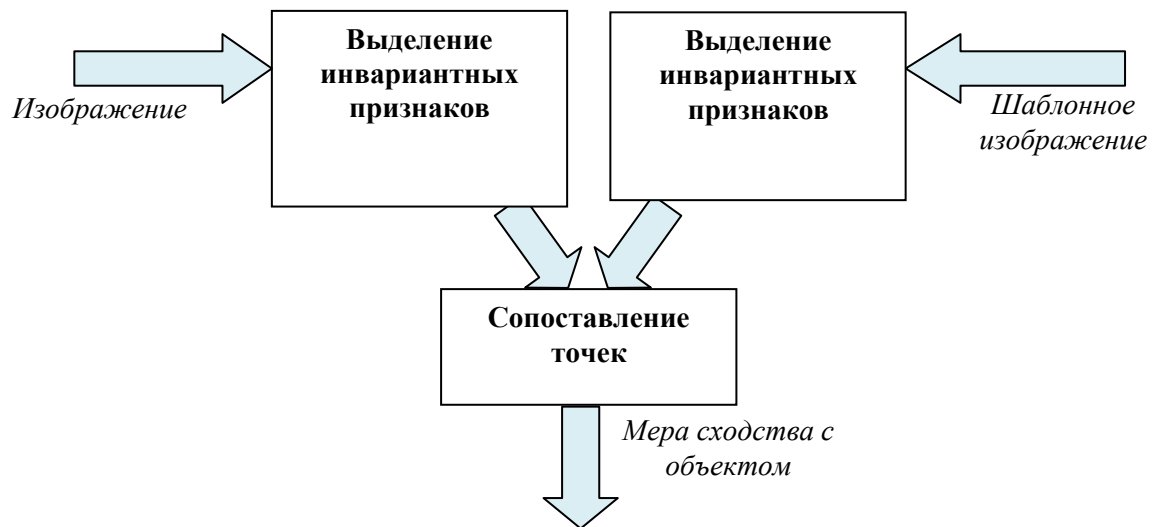


Рисунок 6 – Схема решения задачи детектирования объектов с использованием методов поиска объектов по шаблону

Результатом такого сопоставления (matching) [105] является мера сходства (Рисунок 6). Считается, что если эта мера больше некоторого порога, то тестовое изображение – это изображение объекта.

В простейшем случае в качестве шаблона может выступать изображение объекта – матрица интенсивности цветов, наиболее характерных для объекта. Более сложные методы рассматриваемой группы в качестве шаблона используют наборы векторов признаков (дескрипторы), геометрическое представление объекта [56] или вероятностные модели объектов, которые содержат информацию о распределениях интенсивностей пикселей [5].

В процессе поиска осуществляется проход «бегущим окном», имеющим размеры шаблона, по изображению и сравнение описания части исходного изображения, покрываемого окном, и шаблона. Сопоставление с шаблоном подразумевает сравнение описания тестового и шаблонного изображений по некоторой выбранной метрике [147], как правило, выбирается Евклидово расстояние, норма L_1 , взвешенная свертка квадратичных ошибок, либо корреляция [105].

Допустим, что задано шаблонное описание объекта $I_0(X)$ в дискретном пространстве пикселей $\{X_i = (x_i, y_i)\}$. Тогда задача поиска объекта сводится к задаче минимизации суммарной ошибки. Если в качестве меры сходства использовано Евклидово расстояние, то задача может быть записана следующим образом:

$$E(u) = \sum_i (I(X_i + u) - I_0(X_i))^2 = \sum_i e_i^2 \rightarrow \min,$$

где u – смещение шаблонного описания в системе координат исходного изображения. В конечном итоге, независимо от выбранной метрики приходим к задаче оптимизации.

Для алгоритмов, которые используют сопоставление дескрипторов ключевых точек, одним из наиболее важных является вопрос выбора порога, используемого в качестве критерия соответствия (в простейшем случае, если расстояние между дескрипторами меньше данного порога, точки считаются соответствующими). Увеличение данного порога приводит, с одной стороны, к увеличению числа найденных совпадений (true positives), с другой стороны, к увеличению числа ложных срабатываний (false positives). Уменьшение же порога наряду с ростом числа правильно продетектированных несовпадений (true negatives) ведет к росту числа правильных соответствий, которые были отброшены (false negatives). Данная зависимость графически отображается с помощью ROC-кривой [105] – по величине площади под данной кривой (AUC – area under curve) можно судить о качестве выбранного алгоритма построения соответствий между ключевыми точками на разных изображениях.

Отметим, что методы детектирования по заданному шаблону эффективно работают при поиске одиночных объектов. При возникновении перекрытий в «бегущем окне» исчезают некоторые признаки в описании. Поэтому при сопоставлении окна шаблону вводится порог, по которому отсекаются неперспективные окна – окна, заведомо не содержащие объектов.

5. Литература

5.1. Основная литература

1. Agrawal R., Srikant R. (1994) Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C (eds.) Proc. 20th Int. Conf. on Very Large Data Bases, VLDB'94, Morgan Kaufmann, 487-499.
2. Alkhalid A, Chikalov I, Hussain S, Moshkov M (2011) Extensions of dynamic programming as a new tool for decision tree optimization. In: Ramanna S, Howlett RJ, Jain LC (eds.) Emerging Paradigms in Machine Learning, Springer (to appear).

3. Alkhalid A, Chikalov I, Moshkov M (2010) On algorithm for building of optimal α -decision trees. In: Szczuka MS, Kryszkiewicz M, Ramanna S, Jensen R, Hu Q (eds.) RSCTC 2010. LNCS, vol. 6086:438-445, Springer, Heidelberg.
4. Alonso D., Saldaro L., Nieto M. Robust Vehicle Detection through Multidimensional Classification for On Board Video Based Systems // IEEE. – 2007.
5. Amit Y. 2D Object Detection and Recognition: models, algorithms and networks. – The MIT Press, 2002. – 325p.
6. Andrews S., Tsochantaridis I., Hofmann T. Support vector machines for multiple-instance learning. Advances in Neural Information Processing Systems, 2002. P. 561-568.
7. Apolloni B., Ghosh A., Alpaslan F., Jain L.C., Patnaik S. Machine Learning and Robot Perception. — Springer, 2005.
8. Arrospide J., Salgado L., Nieto M., Jaureguizar F. Robust vehicle detection through multidimensional classification for on board video based systems // ICIP, 2008. – pp. 2008-2011.
9. Arndt R., Schweiger R. Ritter W., Paulus D., Lhlein O. Detection and tracking of multiple pedestrians in automotive applications // Intelligent Vehicles Symposium. 2007. – pp. 13-18.
10. Arth C., Limberger F., Bischof H. Real-Time License Plate Recognition on an Embedded DSP-Platform // Computer Vision and Pattern Recognition, 2007.
11. Aureta L., Aldrich C. Empirical comparison of tree ensemble variable importance measures // Chemometrics and Intelligent Laboratory Systems. Vol.105. No.2, 2011. – pp. 157-170.
12. Ballard D.H., Brown C.M. Computer Vision. – Prentice Hall Inc., 1982. – 539p.

13. Bauer E., Kohavi R. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants // Mach. Learn. 1999. V. 36, №. 1-2., P. 105-139.
14. Bay H., Ess A., Tuytelaars T., Gool L.V. SURF: speed up robust features // Computer Vision and Image Understanding (CVIU), Vol.110, No.3, 2008. – pp. 346-359.
15. Bertozzi M., Broggi A., Fascioli A., Tibaldi A., Chapuis R., Chausse F. Pedestrian localization and tracking system with Kalman filtering // Intelligent Vehicles Symposium. 2004. - pp. 584-589.
16. Binelli E., Broggi A., Fascioli A., Ghidoni S., Grisleri P., Graf T., Meinecke M.-M. A modular tracking system for far infrared pedestrian recognition // Intelligent Vehicles Symposium. 2005. – pp. 759-764.
17. Boryczka U, Kozak J (2009) New algorithms for generation decision trees - Ant-Miner and its modifications. Foundations of Computational Intelligence 6:229-262.
18. Bosch A., Zisserman A., Munoz X. Image classification using random forests and ferns // In 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007.
19. Bradski G., Kaehler A. Learning OpenCV Computer Vision with OpenCV Library. – O' Reilly Media Publishers, 2008. – 571p.
20. Breiman L. Random Forests // Mach. Learn. 2001. V. 45, №. 1, P. 5-32.
21. Breiman L., Friedman J.H., Olshen R.A., Stone C.J. Classification and Regression Trees. Wadsworth & Brooks, 1984.
22. Calonder M., Lepetit V., Strecha C., Fua P. BRIEF: Binary Robust Independent Elementary Features // 11th European Conference on Computer Vision (ECCV), 2010.

23. Chikalov I (2001) Algorithm for constructing of decision trees with minimal number of nodes. In: Ziarko W, Yao YY (eds.) RSCTC 2000, Revised Papers. LNCS, vol. 2005:139-143, Springer, Heidelberg.
24. Chikalov I, Moshkov M, Zielosko B (2011) Online learning algorithm for ensemble of decision rules. In: Kuznetsov SO, Sluzak D, Hepting DH, Mirkin BG (eds.) RSFDGrC 2011. LNCS (LNAI), vol. 6743:310-313, Springer, Heidelberg.
25. Comaniciu D., Ramesh V., Meer P. Real-time tracking of non-rigid objects using mean shift // IEEE CVPR'2000, Vol.2, 2000. – pp. 142-149.
26. Dalal N., Triggs B. Histograms of oriented gradients for human detection // Computer Vision and Pattern Recognition (CVPR'05), 2005.
27. Dance C., Willamowski J., Fan L., Bray C., Csurka G. Visual categorization with bags of keypoints // In: ECCV International Workshop on Statistical Learning in Computer Vision, 2004.
28. Deng J., Berg A., Fei-Fei L. Hierarchical Semantic Indexing for Large Scale Image Retrieval // IEEE Computer Vision and Pattern Recognition (CVPR) – 2011.
29. Deng J., Berg A., Li K., Fei-Fei L. What does classifying more than 10,000 image categories tell us? // Proceedings of the 12th European Conference of Computer Vision (ECCV) – 2010.
30. Deng J., Dong W., Socher R., Li L., Li K., Fei-Fei L. ImageNet: A large-scale hierarchical image database // In CVPR09. – 2009.
31. Dollar P., Belongie S., Perona P. The fastest pedestrian detector in the west // Machine Vision. 2010.
32. Dollar P., Wojek C., Schiele B., Perona P. Pedestrian Detection: An Evaluation of the State of the Art // Pattern Analysis and Machine Intelligence (PAMI'11). 2011.

33. Druzhkov P. N., Eruhimov V. L., Kozinov E. A., Kustikova V. D., Meyerov I. B., Polovinkin A. N., Zolotykh N. Yu. On some new object detection features in OpenCV Library // Pattern Recognition and Image Analysis. 2011. V. 21, № 2. P. 377–379.
34. Duda R.O., Hart P.E., Stork D.G. Pattern classification (2nd edition). — Wiley, 2001.
35. Enzweiler M., Gavrila D. M.. Monocular Pedestrian Detection: Survey and Experiments // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, V. 31, N. 12, P. 2179-2195.
36. Ewens W.J., Grant G. Grant Statistical Methods in Bioinformatics: An Introduction. — Springer, 2005.
37. Exner D., Bruns E., Kurz D., Grundhofer A. Fast and robust CAMShift tracking // IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2010. — pp. 9-16.
38. Fellbaum C. WordNet: An Electronic Lexical Database. — MIT Press, 1998.
39. Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D. Object Detection with Discriminatively Trained Part Based Models // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. Vol.32, No.9. — pp. 1627–1645.
40. Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D. Cascade object detection with deformable path model // Computer Vision and Pattern Recognition (CVPR'10). 2010.
41. Frank A, Asuncion A (2010) UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>. Cited 2 July 2011.
42. Franke U., Joss A. Real-time stereo vision for urban traffic scene understanding // Intelligent Vehicles Symposium. 2000. — pp. 273-278.

43. Freeman W., Adelson E. The design and use of steerable filters // IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.13, No.9, 1991. – pp. 891-906.
44. Freund Y., Schapire R. A decision-theoretic generalization of online learning and an application to boosting // Computer and System Sciences. Vol.55, 1997. – pp. 119-139.
45. Friedman J. Greedy function approximation: the gradient boosting machine // Annals of Statistics. 2001. V. 29, N. 5. P. 1189-1232.
46. Friedman J. H. Greedy Function Approximation: a Gradient Boosting Machine. Technical report. Dept. of Statistics, Stanford University, 1999.
47. Friedman J. H. Stochastic Gradient Boosting. Technical report. Dept. of Statistics, Stanford University, 1999.
48. Friedman J.H., Popescu B.E. Importance Sampled Learning Ensembles // Technical report. Dept. of Statistics, Stanford University, 2003.
49. Gavrila D. M., Giebel J., Munder S. Vision-based pedestrian detection: the protector system // Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy. – 2004. – pp. 13-18.
50. Gavrila D.M. Pedestrian detection from a moving vehicle // Computer Vision. Vol.2. 2000. – pp. 37-49.
51. Geronimo D. A Global Approach to Vision Based Pedestrian Detection for Advanced Driver Assistance Systems, PhD Thesis. Computer Vision Center Barcelona, February 2010.
52. Grauman K., Darrell T. Pyramid match kernels: Discriminative classification with sets of image features // In Proc. ICCV, 2005.
53. Grubb G., Zelinsky A., Nilsson L., Rilbe M. 3D vision sensing for improved pedestrian safety // Intelligent Vehicles Symposium. 2004. – pp. 19-24.

54. Grünwald PD (2007) The Minimum Description Length Principle. Foreword by Jorma Rissanen. Adaptive Computation and Machine Learning. MIT Press.
55. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — Springer, 2001.
56. Hilario C., Collado J.M., Armingol J.M., Escalera A. Pyramidal Image Analysis for Vehicle Detection // Proceedings to Intelligent Vehicles Symposium, 2005. — pp. 88-93.
57. Hirose K., Torio T., Hama H. Robust Extraction of Wheel Region for Vehicle Position Estimation using a Circular Fisheye Camera // International Journal of Computer Science and Network Security, Vol.9, No.12. — 2009.
58. Horn B., Schunk B. Determining Optical Flow // MIT Artificial Intelligence Laboratory, №572. — 1980.
59. Jensen R, Shen Q (2000) Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Transactions on Knowledge and Data Engineering 16:1457-1471.
60. Jin H., Favaro P., Soatto S. Real-time tracking and outlier rejection with changes in illumination // ICCV'01, Vol.1, 2001. — pp. 684-689.
61. Jurafsky D., Martin J.H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Second Edition. — Prentice Hall, 2008.
62. Kalal Z., Mikolajczyk K., Matas J. Forward-backward error: automatic detection of tracking failures // ICPR'10, 2010. — pp. 2756-2759.
63. Kanhere N. K., Pundlik S. J., Birchfield S. T. Vehicle Segmentation and Tracking from a Low-Angle Off-Axis Camera // Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, California, June 2005. – 2005.

64. Ke Y., Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors // Computer Vision and Pattern Recognition (CVPR'04), Vol.2, 2004. – pp. 506-513.

65. Kilian Q. Weinberger, Lawrence K. Saul Distance Metric Learning for Large Margin Nearest Neighbor Classification // JMLR. 2009 V. 10 P. 207-244.

66. Kim J.B., Lee C.W., Lee K.M., Yun T.S., Kim H.J. Wavelet-based Vehicle Tracking for Automatic Traffic Surveillance // IEEE Catalogue No.01CH37239. – 2001.

67. Lazebnik S., Schmid C., Ponce J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories // In: CVPR, 2006.

68. Lee P.H., Chiu T.H., Lin Y.L., Hung Y.P. Real-time pedestrian and vehicle detection in video using 3D cues // ICME'09 Proceedings of the 2009 IEEE international conference on Multimedia and Expo. – 2009. – pp. 614-617.

69. Leibe B., Cornelis N., Cornelis K., Van Gool L. Dynamic 3D scene analysis from a moving vehicle // Computer Vision and Pattern Recognition (CVPR'07). 2007. – pp. 1-8.

70. Leibe B., Leonardis A., Schiele B. Robust Object Detection with Interleaved Object Categoization and Segmentation // Springer Science + Business Media, LLC. – 2007.

71. Leong C.W., Mihalcea R. Measuring the semantic relatedness between words and images // International Conference on Computational Semantics (IWCS) – 2011.

72. Liu B, Abbass HA, McKay B (2003) Classification rule discovery with ant colony optimization. In: Proc. IEEE/WIC Int. Conf. on Intelligent Agent Technology, IAT 2003, IEEE Computer Society, Washington, DC, 83-88.
73. Lowe D. Distinctive image features from scale-invariant keypoints // IJCV 60, P. 91–110, 2004.
74. Lucas B.D., Kanade T. An iterative image registration technique with an application to stereo vision // IJCAI'81, Vol.2, 1981. – pp. 674-679.
75. Matas J., Chum O., Urban M., Pajdla T. Robust wide baseline stereo from maximally stable extremal regions // British Machine Learning Conference, 2002. – pp. 384-393.
76. Michalski SR, Pietrzykowski J (2007) iAQ: A program that discovers rules, AAAI-07 AI Video Competition. In: 22nd AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada.
77. Mikolajczyk K., Schmid C. A Performance Evaluation of Local Descriptors // IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.10, 2005. – pp. 1615-1630.
78. Mikolajczyk K., Schmid C. Scale and affine invariant interest point detectors // International Journal of Computer Vision, 60(1), 2004. – pp. 63-86.
79. Mitchell T. Machine Learning. — McGraw-Hill Science/Engineering/Math, 1997.
80. Moshkov M, Chikalov I (2000) On algorithm for constructing of decision trees with minimal depth. Fundamenta Informaticae 41:295-299.
81. Moshkov M, Piliszcuk M, Zielosko B (2008) Partial Covers, Reducts and Decision Rules in Rough Sets: Theory and Applications. Studies in Computational Intelligence, vol. 145. Springer, Heidelberg.

82. Moshkov M, Zielosko B (2011) Combinatorial Machine Learning: A Rough Set Approach. Studies in Computational Intelligence, vol. 360. Springer, Heidelberg.
83. Munder S., Gavrilă D.M. An experimental study on pedestrian classification // IEEE Pattern Analysis and Machine Intelligence (PAMI'06). Vol.28, No.11, 2006. pp. 1863-1868.
84. Myung Jin Choi, Lim, J.J., Torralba, A., Willsky, A.S. Exploiting Hierarchical Context on a large database of object categories // IEEE Computer Vision and Pattern Recognition (CVPR'10). 2010. – pp. 129-136.
85. Neubeck A., Van Gool L. Efficient Non-Maximum Suppression // International Conference on Pattern Recognition, 2006. – pp. 850-855.
86. Nguyen HS (2006) Approximate Boolean reasoning: foundations and applications in data mining. In: Peters JF, Skowron A (eds.), Transactions on Rough Sets V. LNCS, vol. 4100:334-506, Springer, Heidelberg.
87. Panda B., Herbach J. S., Basu S., Bayardo R. J. PLANET: Massively parallel learning of tree ensembles with MapReduce. Proceedings of the 35th International Conference on Very Large Data Base (VLDB). 2009. P. 1426-1437.
88. Pang J., Huang Q., Jiang S. Multiple instance boost using graph embedding based decision stump for pedestrian detection // Computer Vision. Vol.4, 2008. - pp. 541-552.
89. Papageorgiou C., Poggio T. A trainable system for object detection // Computer Vision. Vol.38. No.1, 2000. – pp. 15-33.
90. Pawlak Z (1991) Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht.

91. Pawlak Z, Skowron A (2007) Rough sets and Boolean reasoning. *Information Sciences* 177: 41-73.
92. Pentland A., Choudhury T. Face Recognition for Smart Environments // *IEEE Computer Vision*. – 2000. – pp. 50-55.
93. Quinlan J.R. Induction of decision trees // *Machine Learning* 1, 1986. – pp. 81-106.
94. Quinlan JR (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo.
95. Rissanen J (1978) Modeling by shortest data description. *Automatica* 14:465-471.
96. Rosten E., Drummond T. Machine Learning for high-speed corner detection // *9th European Conference on Computer Vision (ECCV 2006)*, 2006. – pp. 430-443.
97. Sabzmeydani P., Mori G. Detecting pedestrians by learning shapelet features // *Computer Vision and Pattern Recognition (CVPR'07)*. 2007.
98. Schapire R. The boosting approach to machine learning. An overview // *MSRI workshop on Nonlinear Estimation and Classification*. Springer, 2002.
99. Shi J., Tomasi C. Good features to track // *IEEE*, 1994. – pp. 593-600.
100. Shotton J., Blake A., Cipolla R. Contour-based Learning for Object Detection // *10th IEEE International Conference on Computer Vision (ICCV'05)*. 2005. Vol.1. – P. 503-510.
101. Skowron A (2000) Rough sets in KDD. In: Shi Z, Faltings B, Muslem M. (eds.) *16th World Computer Congress. Proc. Conf. Intelligent Information Processing*, 1-17. Publishing House of Electronic Industry, Beijing.

102. Skowron A, Rauszer C (1992) The discernibility matrices and functions in information systems. In: Slowinski R (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, Kluwer Academic Publishers, Dordrecht, 331-362.
103. Slezak D, Wróblewski J (2003) Order-based genetic algorithms for the search of approximate entropy reducts. In: Wang G, Liu Q, Yao Y, Skowron A (eds.) Proc. 9th Int. Conf. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC 2003. LNAI, vol. 2639:308-311, Springer, Heidelberg.
104. Sonka M., Hlavac V., Boyle R. Image Processing, Analysis and Machine Vision. – Thomson, 2008. – 866p.
105. Szeliski R. Computer Vision: Algorithms and Applications. – Springer, 2010. – 979p.
106. Tamersoy B., Aggarwal J.K Robust Vehicle Detection for Tracking in Highway Surveillance Videos using unsupervised Learning // Advanced Video and Signal Based Surveillance (AVSS '09), 2009. – pp.529-534.
107. Tola E., Lepetit V., Fua P. A Fast Local Descriptor for Dense Matching // IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), 2008. – pp. 1-8.
108. Torralba A., Murphy K.P., Freeman W.T., Rubin M.A. Context-based Vision System for Place and Object Recognition // 9th IEEE International Conference on Computer Vision (ICCV'03). 2003. Vol.1. – pp. 273-283.
109. Tuytelaars T., Mikolajczyk K. Local Invariant Feature Detectors: A Survey // Foundation and Trends in Computer Vision, Vol.3, No. 3, 2007. – pp. 177-280.

110. Tyree S., Weinberger K. Q., Agrawal K., Paykin J. Parallel boosted regression trees for web search ranking. Proceedings of the 20th international conference on World wide web. 2011. P. 387-396.
111. Vasu L., Chandler D.M. Vehicle Tracking Using a Human-Vision-Based Model of Visual Similarity // IEEE. – 2010.
112. Viola P., Jones M., Snow D. Detecting pedestrians using patterns of motion and appearance // Computer Vision. Vol.63. No.2, 2005. – pp. 153-161.
113. Viola P., Jones M.J. Rapid object detection using a boosted cascade of simple features // In Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. – 2001.
114. Viola P., Jones M.J. Robust Real-Time Face Detection // international Journal of Computer Vision 57(2). – 2004. – pp. 137-154.
115. Viola P., Jones M.J., Snow D. Detecting pedestrians using patterns of motion and appearance // In: Proceedings of the 9th International Conference on Computer Vision (ICCV), Vol. 1 – 2003. – pp. 734-741.
116. Walk S., Majer N., Schindler K., Schiele B. New features and insights for pedestrian detection // Computer Vision and Pattern Recognition (CVPR'10). 2010.
117. Wallace CS (2005) Statistical and Inductive Inference by Minimum Message Length. Information Science and Statistics. Springer, New York.
118. Wojek C., Schiele B. A performance evaluation of single and multi-feature people detection // DAGM-Symposium. 2008. – pp.82-91.
119. Wojek C., Walk S., Schiele B. Multi-cue onboard pedestrian detection // Computer Vision and Pattern Recognition (CVPR'09). 2009. – pp. 794 - 801.

120. Wróblewski J (1995) Finding minimal reducts using genetic algorithm. In: Proc. 2nd Annual Join Conf. on Information Sciences, Wrightsville Beach, NC, 186-189.
121. Wu B., Nevatia R. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors // Computer Vision. Vol.75. No.2, 2007. – pp. 247-266.
122. Xu X., Frank E. Logistic regression and boosting for labeled bags of instances // Knowledge Discovery and Data Mining (KDDM'04), 2004.
123. Yike G., Grossman R., eds. High Performance Data Mining: Scaling Algorithms, Applications and Systems. — Kluwer Academic Publishers, 1999.
124. Zhang L., Wu B., Nevatia R. Pedestrian detection in infrared images based on local shape features // Computer Vision and Pattern Recognition (CVPR'07). 2007.
125. Zehang Sun, Bebis G., Miller R. On-road vehicle detection using Gabor filters and support vector machines// Digital Signal Processing, Vol.2, 2002.- pp. 1019-1022.
126. Zhu Q., Avidan S., Yeh M., Cheng K. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients // Computer Vision and Pattern Recognition (CVPR'06). Vol.2, 2006. – pp. 1491-1498.
127. Zielosko B, Moshkov M, Chikalov I (2010) Optimization of decision rules based on methods of dynamic programming. Vestnik of Lobachevsky State University of Nizhni Novgorod 6:195-200 (in Russian)
128. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. Статистические проблемы обучения. М.: Наука, 1974.
129. Дружков П. Н., Золотых Н. Ю., Половинкин А. Н. Параллельная реализация алгоритма предсказания с помощью модели градиентного бустинга деревьев решений // Вестник Южно-

Уральского государственного университета. Серия: Математическое моделирование и программирование. 2011. №. 37 (254). С. 82-89.

130. Дружков П.Н., Золотых Н.Ю., Половинкин А.Н. Программная реализация алгоритма градиентного бустинга деревьев решений. Вестник ННГУ, вып. 1, 2011. С. 193-200.

131. Котов Ю.Б. Новые математические подходы к задачам медицинской диагностики. — М.: УРСС, 2004.

132. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. — М.: Изд. д. Вильямс, 2004. — 465с.

133. Чубукова И. А. Data Mining: учебное пособие. — М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2006.

134. Fergus, R., Perona, P., and Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005), pp. 380–387, San Diego, CA

135. Fergus, R. and Perona, P. and Zisserman, A. Object Class Recognition by Unsupervised Scale-Invariant Learning // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2003, V.2, pp. 264-271

136. Crandall, D., Felzenszwalb, P., and Huttenlocher, D. (2005). Spatial priors for part-based recognition using statistical models. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005), pp. 10–17, San Diego, CA.

137. Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. International Journal of Computer Vision, 61(1):55–79.

138. Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In ECCV

International Workshop on Statistical Learning in Computer Vision,
Prague

139. Bouchard, G. and Triggs, B. (2005). Hierarchical part-based visual object categorization. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005), pp. 709–714, San Diego, CA.

140. Carneiro, G. and Lowe, D. (2006). Sparse flexible models of local features. In Ninth European Conference on Computer Vision (ECCV 2006), pp. 29–43

141. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer, New York, NY.

142. Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. IEEE Transactions on Computers, 22(1):67–92.

143. S. Savarese, J. Winn and A. Criminisi Discriminative Object Class Models of Appearance and Shape by Correlations. In Proc. IEEE Computer Vision and Pattern Recognition (CVPR), New York, 2006.

144. Niebles, J.C., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: Proc. CVPR (2007)

5.2. Ресурсы сети Интернет

145. A summary of vehicle detection and Surveillance Technologies used in Intelligent Transportation Systems
[<http://www.nmsu.edu/~traffic/Publications/Trafficmonitor/vdst.pdf>].

146. Application-Specific Integrated Circuit
[<http://www.siliconfareast.com/asic.htm>].

147. Cai B., Tan F., Lu Y., Zhang D. Knowledge Template Based Multi-perspective Car Recognition Algorithm
[<http://www.mecs-press.org/ijieeb/ijieeb-v2-n2/IJIEEB-V2-N2-6.pdf>].
148. Formal Description of Moravec detector
[<http://kiwi.cs.dal.ca/~dparks/CornerDetection/moravec.htm>].
149. Harris/Plessey Operator
[<http://kiwi.cs.dal.ca/~dparks/CornerDetection/harris.htm>].
150. International Road Dynamics Inc.
[http://www.irdinc.com/products/sensors_accessories/on_road_sensors/].
151. Marsh Products, Inc. [<http://www.marshproducts.com/>].
152. RAI Products [<http://www.raiproductions.com/vehicle-detection-systems.html>].
153. Traffic counting methods
[<http://people.hofstra.edu/geotrans/eng/ch9en/meth9en/ch9m2en.html>].
154. Vaxtor Systems [www.vaxtor.com].
155. ViNotion Inc. [<http://www.vinotion.nl/>].
156. Архив новостей по рубрике Digital Signal Processing
[<http://www.compeljournal.ru/enews/rubric/dsp>].