

Формирование индекса для поиска похожих изображений транспортных средств

©А.Л. Федотов, К.Л. Тассов

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Приведен анализ существующих алгоритмов формирования устойчивых хешей изображений. Предложен алгоритм по формированию индекса для поиска похожих изображений транспортных средств (ТС) на основе найденных особых точек с помощью модифицированного алгоритма SIFT. Для выделения ограничивающего прямоугольника ТС описывается алгоритм, основанный на использовании оси симметрии. Отражены результаты проведенных экспериментов при поиске визуально похожих изображений на основе реальных фотографий ТС. Полученные экспериментальные результаты показывают возможность практического применения предложенного алгоритма в системах идентификации ТС на основе распознавания государственных регистрационных знаков. Предлагаемый алгоритм применим для поиска похожих ТС.

Ключевые слова: *визуально воспринимаемое хеширование, особая точка, дескриптор SIFT, симметрия, транспортное средство, поиск изображений.*

Введение. В настоящее время в развитии цифровых технологий все большую актуальность приобретают задачи создания, обработки, распространения цифровых изображений. Как правило, изображения сохраняются в базах данных. Примером программных комплексов, манипулирующих с такими базами данных, являются системы фото-видеофиксации ТС.

Среди систем видеонаблюдения широко распространены системы идентификации ТС (СИТС), основанные на распознавании государственных регистрационных знаков (ГРЗ) посредством камер видеонаблюдения. СИТС состоит из нескольких модулей считывания (МС), расположенных в различных точках наблюдения, соединенных с центром обработки данных (ЦОД) посредством линии передачи данных. Каждый МС содержит несколько оптико-электронных модулей (ОЭМ) и один сервер распознавания, выполняющий задачи обнаружения события проезда ТС, выделения области ГРЗ и распознавания символов ГРЗ [1].

При идентификации ТС посредством распознавания ГРЗ возможны следующие проблемы: частичная читаемость ГРЗ, ошибки распознавания, умышленная подмена ГРЗ на ТС. В связи с этим становится актуальной задача поиска ТС по марки и модели ТС, в том числе при неизвестном или частично известном ГРЗ.

В настоящий момент система видеонаблюдения проводит распознавание марок по эмблемам [2]. Но этого недостаточно для проведения полноценного поиска, например, угнанных ТС. Поэтому необходимо учитывать модели ТС и проводить поиск похожих ТС. Требуется обеспечить поиск по изображениям, для которых следует формировать индекс в базе данных. Для решения данного вопроса предлагается создать метод, обеспечивающий поиск подобных ТС на изображениях, и его программную реализацию.

В качестве входных данных используются фотографии передней части ТС. Это ограничение связано с конструктивными особенностями расположения ОЭМ над дорожным полотном, позволяющих распознать регистрационные знаки.

Ограничения связаны с обработкой изображений только с одним ТС, а также с качеством фотографий, полученных ОЭМ. Так фотографии, сделанные в светлое и темное время суток, могут отличаться друг от друга, что приведет к ложному сопоставлению. Кроме того, на качество фотографий влияют погодные условия, что может привести к ложным срабатываниям предлагаемого алгоритма.

Особенности построения индекса изображения. При формировании индекса необходимо учитывать визуальную схожесть изображений, а также возможность последующего поиска в базе данных. Для этого по изображению строится хеш. Он должен быть нечувствителен к визуально неизменным областям изображения. Формирование такого вида хеша называется визуально восприимчивым хешированием изображения (англ. perceptual image hashing). Процесс хеширования заключается в получении значения, размер которого намного меньше, чем размер самого изображения, и сохраняющего его семантику. В общем случае хеш возможно представить в виде бинарного кода — функции, выбранной с целью уменьшения размерности пространства

$$y_i = f(x_i) = [h_1(x_i), h_2(x_i), \dots, h_k(x_i)]. \quad (1)$$

Формально требования к изображению и его хеш-функции возможно выразить следующим образом. Пусть для данного изображения I и множества визуально похожих изображений $J = \{J_k, k = \overline{1, N}, N \in \mathbb{N}\}$ существует хеш-функция H . Тогда:

1) хеш-функция представляет собой произвольное отображение, не являющееся взаимно-однозначной функцией: $I \rightarrow H(I)$;

2) размер значения хеш-функции намного меньше, чем размер изображения:

$$|H(I)| \ll |I|;$$

3) визуально похожие изображения имеют одинаковые значения хеш-функций, вероятность этого близка к 1:

$$P(H(I) \approx H(J_k)) \geq 1 - \varepsilon, 0 \leq \varepsilon < 1, k = \overline{1, N};$$

4) визуально отличающиеся изображения имеют различные значения хеш-функций, вероятность близка к 1:

$$P(H(I) \neq H(I')) \geq 1 - \delta, 0 \leq \delta < 1, I' \notin \{J_k\},$$

значения параметров ε и δ близки к 0.

Предлагаемый алгоритм индексирования основывается на определении особых точек изображения. Устойчивость значения хеш-функции зависит от устойчивости алгоритма определения особых точек и метода сжатия полученных дескрипторов в короткий индекс. Концептуально процесс формирования индекса изображения представлен на рис. 1.

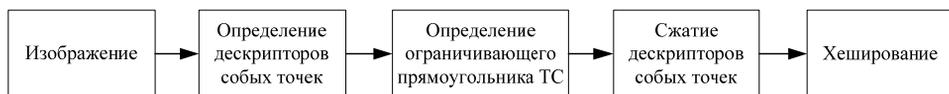


Рис. 1. Формирование значения хеш-вектора

Анализ алгоритмов получения хеш-функций. Существуют несколько классов алгоритмов получения хеш-функций изображений, которые возможно применить к предметной области.

К классу методов визуально восприимчивого хеширования (англ. perceptual image hashing) относят алгоритмы, привязывающиеся к локальным особенностям изображений (например, алгоритмы, основанные на дискретном преобразовании Фурье), а также к особым точкам [3]. Методы, относящиеся к данному классу, устойчивы к аффинным преобразованиям и зашумлениям, однако устойчивость зависит от выбора алгоритма определения особых точек.

К классу методов хеширования изображений по содержанию (англ. content based hashing) относят методы, выявляющие из изображения признаки, основанные на статистических данных интенсивностей участков изображения, таких, как среднее, дисперсия и моменты более высокого порядка [4, 5]. К недостаткам следует отнести то, что методы устойчивы не ко всем аффинным преобразованиям.

Выделяют также класс методов, основанных на обучении (алгоритм усиления простого чувствительного кодирования (англ. Boosting Similarity Sensitive Coding, BoostSSC) [6], ограниченный метод Больцмана (англ. Restricted Boltzmann Machines, RBM)). Методы основываются на использовании нейронных сетей применительно к

вычислению результирующей хеш-функции. Недостатками методов являются сложность настройки нейронных сетей (необходимо сформировать обучающую выборку) и ограниченность применения.

Таким образом, был выбран метод визуально восприимчивого хеширования, основывающийся на выделении особых точек.

Для выделения из изображения некоторой структурированной, осмысленной информации необходимо привязаться к локальным особенностям изображения. Можно выделить особые точки на изображении. Особая точка m , или характеристическая точка (англ. feature point, key point) изображения, — это точка, окрестность которой $o(m)$ можно отличить от окрестности любой другой точки изображения $o(n)$ в некоторой другой окрестности $o_2(m)$ [7]. Для выделения особых точек используется детектор, а для формирования инвариантной структуры — дескриптор.

Детектор — метод определения особых точек из изображения. Детектор обеспечивает инвариантность нахождения одних и тех же особенностей относительно преобразований изображений.

Дескриптор — описатель особой точки, выделяющий ее из остального множества особых точек. В свою очередь, дескрипторы должны обеспечивать инвариантность нахождения соответствия между особыми точками относительно преобразований изображений [8].

Выбор алгоритма вычисления особых точек является важным, поскольку от него зависит устойчивость поиска похожих изображений в базе данных.

Среди существующих алгоритмов особых точек наибольшую устойчивость показывает алгоритм SIFT (англ. Scale Invariant Feature Transform) [9 — 11]. Особые точки представляются структурами в виде капель (англ. blobs) — это круговые области, описываемые четырьмя параметрами: координатами центра (x, y) , масштабом (σ) и направлением (θ) , угол в радианах). Капли имеют непостоянную границу, и их границы могут не совпасть на другом изображении.

Предлагаемый алгоритм. Выделение особых точек. Для обнаружения особых точек используется алгоритм преобразования масштабно-инвариантных характеристик (SIFT) [12]. Он строит пирамиду изображений, применяя фильтр Гаусса, после чего берется разность пирамиды гауссианов (рис. 2). Разностью гауссианов (англ. Difference of Gaussian, DoG) называют изображение, полученное путем последовательного вычитания пикселей одного гауссиана исходного изображения из гауссиана с другим радиусом размытия:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \cdot I(x, y), \quad (2)$$

где D — разность гауссианов; G — гауссово ядро (гауссова функция); k — коэффициент, на который отличаются радиусы размытия σ .

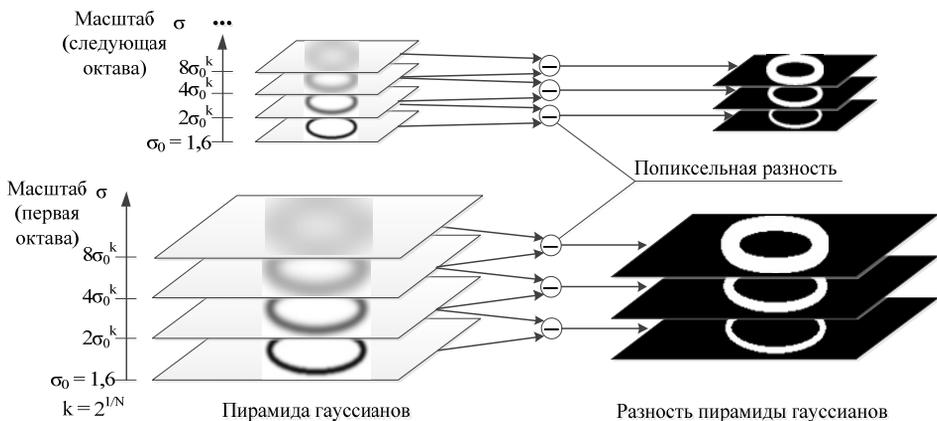


Рис. 2. Процесс построения гауссианов

Далее определяются экстремумы — набор потенциальных особых точек, после чего происходит их уточнение: исключаются точки с малой контрастностью (с помощью вычисления экстремума разности гауссианов) и граничные точки (области, имеющие большой локальный изгиб вдоль границы и малый — в перпендикулярном направлении). Далее для каждой вычисленной особой точки находится ее ориентация. Этот шаг обеспечивает инвариантность к повороту. На конечном шаге строятся дескрипторы. Полученный в результате отбора дескриптор устойчив к перспективным преобразованиям и изменениям освещенности. В результате работы алгоритма на выходе возвращается список особых точек p_i с дескрипторами d_i (формула (3), рис. 4, а):

$$P = \left\{ \left(p_i(x, y, \sigma, \theta), d_i \right) \right\}, i = \overline{1, N}. \quad (3)$$

Определение ТС. Процесс выделения ограничивающего прямоугольника ТС состоит из определения оси симметрии ТС на изображении, выделения ограничивающего прямоугольника и определения центра ТС.

Определение оси симметрии основано на использовании особых точек алгоритмом *SIFT* (3). Для каждого дескриптора особой точки d_i определяется отраженный \bar{d}_i вокруг оси y . Выбор оси отражения является произвольным в силу инвариантности дескриптора d_i .

Дескрипторы особых точек с вычисленными отраженными дескрипторами попарно сравниваются между собой. В результате формируется матрица схожести $D_{NN} = (d_{ij}) : d_{ij} = d(d_i, \bar{d}_j) \forall p_i, p_j \in P, i, j = \overline{1, N}, i < j$, где $d(d_i, \bar{d}_j)$ — евклидово расстояние между элементами дескрипторов d_i и \bar{d}_j . Полученная матрица является симметриче-

ской, поскольку каждая пара особых точек формирует два одинаковых элемента. Далее в каждой строке матрицы D определяются минимальный элемент и его индекс: $k_i = \arg \min_j d_{ij}, i, j = \overline{1, N}, i < j$. Таким образом, формируется множество потенциальных сопоставлений особых точек $M = \{(p_i, p_j) : j = k_i, i = \overline{1, N}\}$. Полученные пары сопоставленных дескрипторов проходят уточнение по углам и масштабу (рис. 3) [13].

Вначале вычисляется угловая весовая функция $\Phi_{ij} \in [-1; 1]$:

$$\Phi_{ij} = -\cos(\theta_i + \theta_j - 2\varphi_{i,j}). \quad (4)$$

Если значение $\Phi_{ij} < 0$, то пара особых точек исключаются из дальнейшего рассмотрения как несимметричные.

Затем вычисляется мера масштабирования $S_{ij} \in [0; 1]$, в котором найдены особые точки. Данная мера позволит отбросить точки, найденные в разных масштабах, но имеющие похожие дескрипторы:

$$S_{ij} = \frac{|\sigma_i - \sigma_j|}{\max(\sigma_i, \sigma_j)} \quad (5)$$

Если значение $S_{ij} < \Delta = 0,3$, то пара особых точек исключаются из дальнейшего рассмотрения как несимметричные.

Для оставшихся пар симметричных особых точек вычисляется общий параметр симметрии $M_{ij} \in [0; 1]$: $M_{ij} = \Phi_{ij} S_{ij}, i, j = \overline{1, N_M}$, где N_M — количество особых точек, оставшихся после уточнения ($N_M \leq N$). Каждая пара особых точек определяет потенциальную ось симметрии, проходящую перпендикулярно через центр C_{ij} отрезка, соединяющего парные особые точки (пунктирная линия на рис. 3). Пример найденных потенциальных осей симметрии показан на рис. 4, б).

Все потенциальные линии осей симметрии представляются в полярных координатах в виде

$$r_{ij} = x_c \cos \varphi_{ij} + y_c \sin \varphi_{ij}, \quad (6)$$

где (x_c, y_c) — центр середины отрезка C_{ij} ; φ_{ij} — угол наклона отрезка, соединяющего парные особые точки.

Для поиска преобладающей оси симметрии используется линейное преобразование Хафа: каждая середина (x_c, y_c) пары особых точек преобразуется в пространство Хафа (r_{ij}, φ_{ij}) со значением параметра симметрии M_{ij} . Результирующее пространство Хафа размывается фильтром Гаусса. Точка, имеющая максимальное значение координаты симметрии $(r_{\max}, \varphi_{\max})$, берется за основную ось симметрии. Соответ-

ствующие координаты в декартовой системе координат $C_s(x_s, y_s)$ станут центром оси симметрии. Точки, лежащие в окрестности найденной особой точки, будут симметричными, т. е. выбираются особые точки $p(r_i, \varphi_i) : d(p(r_{\max}, \varphi_{\max}), p(r_i, \varphi_i)) \leq \delta, i = \overline{1, N_M}$ (рис. 4, в)). Параметр δ задает окрестность, в которой рассматриваются симметричные точки. Таким образом, сформируется M особых точек. Длина оси симметрии определяется как максимальное расстояние между найденными симметричными точками. В случае если угол наклона оси симметрии проходит более чем 30° относительно оси y , то ось симметрии исключается из рассмотрения как заведомо ложное срабатывание (связано с конструктивными особенностями размещения камер видеорегистрации).

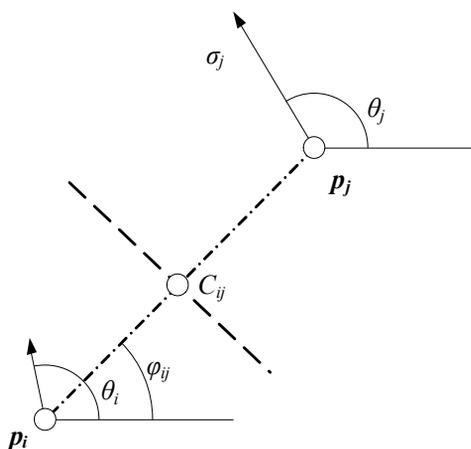


Рис. 3. Определение потенциальной оси симметрии для пары особых точек



Рис. 4. Этапы определения ограничивающего прямоугольника ТС

Центр изображения, относительно которого производится хеширование, находится на оси симметрии. Изображение и координаты особых точек поворачиваются на угол φ_{\max} для совмещения оси симметрии с осью y .

Для получения ограничивающего прямоугольника R используется метод скользящего окна. Для определения левой и правой границ пря-

моугольника формируется скользящее окно, имеющее размеры $20 \times H_s$ пикселей, где H_s — высота оси симметрии. Последовательно перемещающаяся плавающая окно влево и вправо (для левой и правой частей относительно оси симметрии соответственно), подсчитывают количество особых точек, попавших в окно. Выбирают то окно, в котором количество особых точек k_x максимально. В силу особенности алгоритма определения особых точек, они располагаются на автомобилях, в частности на границах ТС. Длина прямоугольника (левое x_l и правое x_r смещения по оси x) рассчитывается по следующим формулам:

$$k_x \rightarrow \max, x_l = x_p^l - x, x_r = x_p^r + x, x \in [1; \dots; \min(W - x_p^r; x_p^l)], \text{ если}$$

ТС целиком помещается в окно;

$$k_x \rightarrow \max, x_l = x_p^l - x, x_r = W, x \in [1; \dots; x_p^l], \text{ если левая часть ТС}$$

помещается в окно, а правая обрезана;

$$k_x \rightarrow \max, x_l = 1, x_r = x_p^r + x, x \in [1; \dots; W - x_p^r], \text{ если правая часть ТС}$$

помещается в окно, а левая обрезана;

$$x_l = 1, x_r = W, \text{ если ТС полностью помещается в изображение.}$$

Здесь W — длина изображения; x_p^l и x_p^r — левая и правая координаты по оси x самой отдаленной точки от центра симметрии x_{\max} .

Высота прямоугольника (верхнее y_t и нижнее y_b смещения по оси y) определяется следующим образом. Ось симметрии располагается на нижней части ТС, поскольку количество особых точек в этой части самое большое в ТС. Нижняя часть ТС фиксируется постоянно в силу работы СИТС и камер видеорегистрации. Значит, нижняя часть ТС — это $y_b = \min_y p_i, i = \overline{1, M}$, где p_i — особые точки, принадлежащие оси симметрии. Для определения верхней части используется метод плавающего окна, имеющего размеры $w \times 20$ пикселей, движущегося вверх, начиная с $\min_y p_i + w$, где $w = x_r - r_l$ — длина ограничивающего прямоугольника (предполагается, что высота ТС не может быть меньше половины его длины). Выбирается то окно, в котором количество особых точек k_y максимально. Смещение по оси y верхней части y_t прямоугольника

$$k_y \rightarrow \max, y_t = y_b + w/2 + y, y \in [1; \dots; H - y_p^b - w/2],$$

если верхняя часть ТС помещается в окно;

$$y_t = W,$$

если верхняя часть ТС не помещается в окно.

Случай, когда $h < w/2$, не рассматривается (т. е. видна лишь нижняя часть ТС с областью ГРЗ). Таким образом, центром изображения ТС будет являться точка $C_V(x_c, y_c)$, лежащая внутри ограничивающе-

го прямоугольника R : $x_c = x_s, y_c = y_b + h/2$, где x_s — координата по оси x центра симметрии; y_b — нижнее смещение по оси y ограничивающего прямоугольника; h — высота ограничивающего прямоугольника (рис. 4, z).

Формирование хеш-вектора. Сформированные дескрипторы особых точек представляют собой описатели, содержащие достаточную информацию для их идентификации и сравнения на парах изображений. Однако количество особых точек на одном изображении 384×272 исчисляется сотнями, что делает исчерпывающий поиск неприменимым, по крайней мере, по двум причинам. Во-первых, для каждого изображения необходимо хранить структуру, содержащую все найденные особые точки, размер которой превышает размер самого изображения. Во-вторых, время поиска не будет отвечать поисковому запросу, поскольку изображения в базе данных исчисляются миллионами.

Поэтому необходим алгоритм, сводящий все найденные особые точки внутри ограничивающего прямоугольника R и их дескрипторы в одну структуру, имеющую меньшую размерность, которая и будет являться хешем с необходимой длиной L : $H = [h_1, h_2, \dots, h_L]$.

Центральная часть изображения в общем случае менее подвержена искажениям и преобразованиям, поэтому центральная точка принимается равной центру выделенного ТС $C_V = (x_c, y_c)$. Пусть получено множество особых точек $P \in R$ (3). Выделим особые точки, лежащие в секторах $b_k, k = \overline{1, L}$, выходящих из центральной точки C_V с углом, равным $\varphi = 2\pi/L$ (рис. 5, формула (7)):

$$b_k = \left\{ p_i \in P : (k-1)l \geq (\theta_{p_i} - \theta_C) < kl \right\}, \quad (7)$$

где $(\theta_{p_i} - \theta_C) = \Delta\theta_{p_i} \in [0; 2\pi)$ — относительная угловая разность между направлением особой p_i и центральной C точками. Угол θ_C определяется равным φ_{\max} .

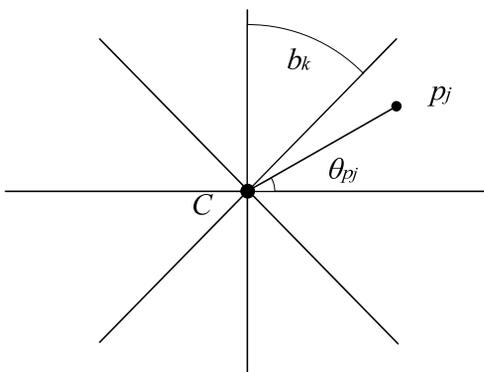


Рис. 5. Процесс хеширования

Сгенерируем случайные веса по нормальному распределению $\{a_k\} \in N(\mu, \sigma^2)$, $k = \overline{1, L}$. Каждый вес α_k — это 128-мерный вектор, совпадающий с размерностью дескриптора, полученного алгоритмом SIFT. Сгенерируем случайные веса по равномерному распределению $W = \{w_k\} \in U(0.5, 1)$, $k = \overline{1, L}$. Данные веса необходимы для различия точек, располагающихся на разных расстояниях от центральной, но лежащие в одном секторе b_k на одном и том же луче.

Вычислим каждый компонент хеш-вектора следующим способом:

$$h_k = \sum_{p_i \in b_k} w_{\left\lfloor \frac{L \|p_i - C_V\|}{\|C_V\|} \right\rfloor} (\alpha_k, d_{p_i}), \quad (8)$$

где $\|p_i - C_V\|$ — норма вектора, образованного разностью координат особой и центральной точек; $\|C_V\|$ — норма вектора, образованного разностью центральной точки и центра координат.

Случайный выбор весов W может негативно повлиять на конечное значение хеш-вектора. Поэтому сгенерированные веса предварительно сортируются в соответствии со значением (α_k, d_{p_i}) . В результате элементы, имеющие большие значения, перемножаются с бóльшими по значению весами.

Полученный хеш-вектор H состоит из L чисел с плавающей запятой (типа *double*, занимающий 8 Байт). Для сокращения размера каждый компонент хеша приводится к целому числу, занимающему 1 Байт:

$$h'_k = \left\lfloor \frac{h_k}{\max} \cdot 256 \right\rfloor, \quad h'_k \in \{0, \dots, 255\},$$

где \max — максимальное значение, возможное для компонента вектора.

Выбор длины L хеш-вектора зависит от того, насколько точным должен быть поиск изображений. Чем больше длина вектора, тем точнее производится поиск, и наоборот. В данной работе длина хеш-вектора принята равной 20, основываясь на исследованиях в работах [14–16], а также по результатам проведенных экспериментов.

Осуществление поиска в базе данных. По полученному хеш-вектору возможно производить поиск, последовательно перебрав по всей базе данных и сравнив со всеми полученными хеш-векторами, что делает подобную операцию невыполнимой с точки зрения времени отклика системы в масштабах большой выборки из миллионов изображений. Поэтому необходимо сформировать атрибут из хеш-вектора, по которому возможно проводить индексирование. Для этого необходимо преобразовать хеш-вектор в индекс, используя

нейронную сеть, основанную на двух сетях Кохонена, кластеризующую входные данные. Обучающее множество состоит из входных векторов. На вход сети поступает множество многомерных векторов $H^i = [h_1, h_2, \dots, h_L]$.

Первый слой состоит из 20 нейронов. Количество выходных классов будет соответствовать степени схожести хеш-векторов, а значит, схожести ТС. Второй слой представляет собой совокупность нейронных сетей Кохонена, динамически формируемых в результате обучения первого слоя. Количество нейронов в выходном слое равно $M = 65536$ (2 Байт). Каждое число представляет собой группу схожих изображений $\{C^m\}$, $m = \overline{1, M}$, т. е. определяются ядра классов $\{C^m\}$, $m = \overline{1, M}$ в пространстве классов C так, чтобы меры близости $d(H^i, c^m)$ были минимальны: $\sum_p d(H^i, c^{m(i)}) \rightarrow \min$, где $m(i)$ — функция, которая определяет номер класса по индексу p множества хеш-векторов $\{H^i\}$, задает разбиение на классы и является решением задачи кластеризации. Ядра c^m являются весовыми коэффициентами нейронов. Каждый нейрон сети Кохонена запоминает один класс, т. е. величина выхода тем выше, чем ближе вектор к данному классу.

На выходном слое определяется наиболее схожий нейрон, у которого скалярное произведение весов и поданного на вход вектора максимально. Вычисленный нейрон объявляется победителем (англ. winner или best match unit, ВМУ), и происходит изменение направления вектора к максимально подходящему кластеру.

При обучении сеть подстраивается под закономерности входных данных. Обучение начинается с инициализации весовых коэффициентов значениями, равными $1/\sqrt{L}$, где L — длина хеш-вектора.

Таким образом, индекс строится средствами СУБД по числу длины 4 Байт (тип *int* на примере СУБД *Microsoft SQL Server 2012*), причем в последнем числе будут использованы только младшие 2 Байт (старшие 2 Байт оставить зарезервированными).

Перед подачей хеш-вектора в сеть происходит предобработка данных:

- определение по каждой компоненте хеш-вектора среднего значения;
- смещение начала координат в среднюю точку;
- нормализация полученных векторов.

Результаты проведения экспериментов. Для оценки работы алгоритма выбраны $N = 10000$ фотографий различных ТС, зафиксированных фронтальной стороной к камере видеорегистрации, полученных в формате JPEG, 384×272 . Выбраны модели четырех марок ТС (по 2500 изображений в группе): Форд, Ауди, Митсубиси, Мерседес-Бенц.

Результаты поиска запрашиваемых изображений автомобилей из БД показаны в таблице. Для проведения эксперимента выбраны две гипотезы: правильно найденные изображения H_0 и неверно найденные изображения H_1 . Пороговое значение метрики принято 55 с использованием евклидовой нормы. Проверка результатов проводилась экспертным образом. Также определены ложноположительные (ошибка первого рода) и ложноотрицательные результаты (ошибка второго рода). Ложноположительными результатами считаются найденные изображения, не соответствующие поисковому запросу. Данные сгруппированы по маркам автомобилей.

Результаты обработки 10000 изображений

Результаты	Форд	Ауди	Митсубиси	Мерседес-Бенц
Положительные	65,52%	66,67%	59,14%	79,12%
Отрицательные	94,37%	90,43%	93,02%	86,67%
Ложноположительные	34,48%	33,33%	40,86%	20,88%
Ложноотрицательные	5,63%	9,57%	6,98%	13,33%

Также проведены эксперименты сравнений хеш-векторов длиной 12, 16, 24, 28 и 32 Байт. Наилучший результат показали сравнения с векторами длиной 20, что совпадает с результатами исследований в работах [14–16].

Заключение. Проведен анализ предметной области для получения устойчивых хеш-функций изображений. Разработан алгоритм определения ТС, используя ось симметрии. Выделен алгоритм формирования многомерного индекса для поиска похожих изображений, а также — среди существующих в базе данных.

Экспериментально полученные результаты показывают возможность практического применения выбранного алгоритма в системах идентификации ТС на основе распознавания ГРЗ.

Для дальнейшего развития следует использовать методы хеширования изображений по содержанию (*content based hashing*) для обеспечения устойчивости поиска, такие как LSH, NFM, FJLT, отличные от предложенного метода визуально восприимчивого хеширования. В данном классе методов применяется уменьшение пространства за счет случайной выборки подизображений и не используется информация об особых точках. Поскольку происходит произвольная выборка подизображений, данный способ позволит сохранить информацию об изображении в целом, даже если часть изображения была потеряна. Также возможно использование других методов выявления особых точек, более устойчивых к световым бликам и к зашумлению. Использование комбинированного подхода позволит совместить достоинства двух методов.

ЛИТЕРАТУРА

- [1] Тассов К.Л. Методы предварительной обработки изображения в системе идентификации транспортных средств. *Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение»*, 2012.
- [2] Копылов И.Е., Тассов К.Л. Определение марки автомобиля по видеокadres с использованием модифицированного алгоритма Виолы — Джонса. *Наука и образование: электронное научно-техническое издание*. Москва, МГТУ им. Н.Э. Баумана, 2012, № 6, 13 с.
- [3] Monga V., Mihcak M.K. Robust and secure image hashing via nonnegative matrix factorizations. *Information Forensics and Security, IEEE Trans on*, 2007, vol. 2, issue 3, pp. 376–390.
- [4] Kailasanathan C., Naini R.S. Image authentication surviving acceptable modifications using statistical measures and k-mean segmentation. *IEEE-EURASIP Work. Nonlinear Sig. and Image Processing*, 2001, vol. 1.
- [5] Venkatesan R., Koon S.M. Robust image hashing. *Proc. IEEE Conf. on Image Processing*, 2000, vol. 3, pp. 664–666.
- [6] Fast pose estimation with parameter sensitive hashing. *ICCV*, 2003.
- [7] Конушин А.С. Слежение за точечными особенностями сцены (Point feature tracking). *Компьютерная графика и мультимедиа*, 2003, № 1(5).
- [8] Гонсалес Р., Вудс Р. *Цифровая обработка изображений*. Москва, Техносфера, 2005, 1072 с.
- [9] Juan L, Gwun O. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing IJIP*, 2009, vol. 3, no. 4, pp. 143–152.
- [10] Valgren C., Lilienthal A.J. SIFT, SURF & Seasons: Appearance-based Long-term Localization in Outdoor Environments. *AASS Research Centre, Dept. of Computer Science, Orebro University*. Orebro, Sweden, 2010.
- [11] A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 2005, vol. 65, no. 1–2, pp. 43–72.
- [12] Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 2004, vol. 60, no. 2, pp. 91–110.
- [13] Loy G., Eklundh J.-O. Detecting symmetry and symmetric constellations of features. *Proceedings of the 9th European conference on Computer Vision*, 2006, vol. II, pp. 508–521.
- [14] Xudong Lv., Wang Z. Jane. An extended image hashing concept: content-based fingerprinting using FJLT. *EURASIP J. Inf. Secur.* Article 2, 2009, pp. 1–16.
- [15] Monga V., Evans B.L. Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs. *Trans. Img. Proc.* 2006, vol. 15, no. 11, pp. 3452–3465.
- [16] An efficient parts-based near-duplicate and sub-image retrieval system. *Proceedings of the 12th annual ACM international conference on Multimedia*, 2004, pp. 869–876.

Статья поступила в редакцию 10.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Федотов А.Л., Тассов К.Л. Формирование индекса для поиска похожих изображений транспортных средств. *Инженерный журнал: наука и инновации*, 2013, вып. 6. URL: <http://engjournal.ru/catalog/it/hidden/1100.html>

Федотов Андрей Леонидович родился в 1988 г., студент кафедры «Программное обеспечение ЭВМ и информационные технологии» факультета «Информатика и

системы управления» МГТУ им. Н.Э. Баумана с 2007 г. Область научных интересов: цифровая обработка данных. e-mail: fedotovand@mail.ru

Тассов Кирилл Леонидович родился в 1966 г., окончил МГТУ им. Н.Э. Баумана в 1991 г. Старший преподаватель кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана. Автор научных работ в области теории распознавания образов и цифровой обработки сигналов. e-mail: ktassov@policesoft.ru