

Аннотирование изображений электронной коллекции исторических фотографий

А. Н. Талбонен

Петрозаводский государственный университет (ПетрГУ),

Петрозаводск

perhetal@onego.ru

А. А. Рогов

rogov@psu.karelia.ru

Аннотация

В данной статье рассматриваются методы аннотирования изображений цифровых исторических коллекций за счет наличия лиц с помощью эксперта и за счет текстур с помощью метода моментов. Предлагается улучшенный метод обнаружения лиц, основанный на комбинации алгоритма Виолы-Джонса и алгоритма локальных бинарных шаблонов.

1 Введение

На данный момент существует множество различных методов автоматического аннотирования изображений. Все такие методы можно разделить на 2 категории: алгоритмы аннотирования по всему изображению и алгоритмы аннотирования по отдельным объектам на изображении. В алгоритмах аннотирования по объектам, будем называть изображением участок данного изображения, содержащего искомый объект, а также изображение данного объекта, используемое для обучения. Большинство алгоритмов аннотирования работают по одной общей схеме:

1. Выделение набора признаков для изображения или объекта.
2. Подбор коллекции изображений и обучение классификатора.
3. Классификация коллекции изображений, в результате которой каждому изображению будет соответствовать набор меток.

Методам поиска объектов на изображениях, включая методы извлечения текстовых меток, посвящено множество исследований. Известный семинар РОМИП собирает множество различных работ по поиску и аннотированию изображений. К примеру, в статьях [4, 5] описаны методы решения однотипных задач поиска изображений по содержанию: поиска нечетких дубликатов, поиска изображений по образцу, которые рассматривались на семинаре РОМИП-2010. Методы поиска нечетких дубликатов и поиска изображений по образцу, описанные в данных статьях обладают наибольшими оценками полноты и точности [1], среди других методов, представленных на семинаре. Кроме того, в статье [4]

описан метод построения текстовых меток для изображений, основанный на поиске нечетких дубликатов для данного изображения в заранее аннотированной коллекции большого объема.

В данной работе рассматриваются методы аннотирования на основе текстурных характеристик и наличия лиц людей на изображении. Анализ методов аннотирования проводился на коллекции исторических фотографий со строительства Беломорско-Балтийского канала и альбома Бродаца [7]. Особенностью исследования является работа с изображениями низкого качества, что свойственно коллекциям исторических фотографий.

2 Аннотирование на основе наличия лиц на изображениях

2.1 Особенности задачи

Исходными данными для проведенных исследований является электронная коллекция черно-белых изображений строительства Беломорско-Балтийского канала (ББК) в формате JPEG, созданных около 80 лет назад. Коллекция насчитывает более 6 тыс. изображений. Преобладание диапазона лиц размера 20–40 пикселей является следствием низкого качества, как оригинальных фотографий, так и оцифрованных изображений.

Для алгоритмов детектирования характерны ложные срабатывания, когда определенный участок изображения, не являющийся лицом, но содержащий некоторые локальные признаки, признается классификатором как лицо. Как следствие, снижается точность результатов. При этом наличие большого количества деталей и шумов повышает вероятность ложного срабатывания. Изменяя параметры алгоритма можно оказывать влияние на вероятность обнаружения лица или ложного срабатывания. Но при этом в рамках одного алгоритма можно повысить точность за счет полноты и наоборот. Таким образом, можно наблюдать некоторую обратную зависимость, не позволяющую повысить оба показателя. Для решения этой проблемы можно использовать дополнительную классификацию: сортировку результатов детектирования с наибольшей полнотой.

2.2 Общая схема аннотирования

В данной работе исследуется случай, когда отсутствует какая-либо база данных лиц, необходимая для автоматического распознавания лиц на изображениях. В подобных условиях работа по аннотированию возлагается на эксперта. Однако при наличии вспомогательных средств можно автоматизировать данный процесс.

В данной статье предлагается следующий метод аннотирования:

1. Извлечение объектов (лиц).
2. Расчет попарных расстояний между объектами. Данный процесс имеет сложность выделения памяти $O(N^2)$, поэтому для экономии можно сохранять расстояния между несколькими ближайшими объектами, например, 10.
3. Аннотирование. Данный процесс является итеративным и может выполняться до тех пор, пока все найденные объекты не будут помечены. Итерация включает в себя следующее:
 - a. Эксперт выбирает очередной непомеченный объект, после чего система предлагает выбрать один из ближайших объектов в порядке возрастания расстояния до текущего объекта.
 - b. Если эксперт обнаруживает сходство между объектами, система создает связь между ними.
 - c. Если текущий объект и связанные с ним объекты не имеют метки, эксперт помечает данные объекты: в качестве метки может выступать ФИО. После этого система присваивает введенную метку всем связанным объектам и далее они считаются помеченными.

2.3 Обзор методов извлечения лиц

На данный момент существует множество алгоритмов распознавания лиц и соответствующих коммерчески успешных систем, например, Microsoft Kinect. Среди бесплатных систем, позволяющих распознавать лица, можно выделить Google Picasa [16]. Однако в обоих случаях алгоритмы распознавания являются закрытыми и не могут быть использованы в свободных проектах.

В данной работе для обнаружения лиц используется алгоритм Виолы-Джонса [18]. Данный алгоритм был выбран, потому что он обладает высокими показателями по сравнению с другими бесплатными / широкодоступными алгоритмами [9, 19].

Широко распространенная библиотека openCV содержит реализацию данного алгоритма на основе каскадных классификаторов в методе detectMultiScale класса cv::CascadeClassifier [8]. Изменяя параметры данного метода можно получать результаты поиска с определенной полнотой и точностью. Кроме того, метод detectMultiScale

позволяет только обнаруживать объекты лиц, но никак не сравнивать их между собой. Поэтому для аннотирования изображений с помощью поиска лиц метод detectMultiScale было решено дополнить другим методом, позволяющим сравнивать изображения лиц между собой.

Некоторые программы, например, Google Picasa, позволяют искать похожие лица, однако это эффективно только для объектов с большими размерами.

В работах [2, 6] было предложено использовать локальные бинарные шаблоны (ЛБШ, LBP) для распознавания лиц. В данной работе предлагается использовать LBP для повышения точности и полноты метода detectMultiScale.

2.4 Распознавание объектов с помощью LBP

LBP, впервые описанный в [14], представляет собой простой и мощный инструмент для распознавания различных элементов изображения. В качестве пространства признаков используются гистограммы так называемых кодов LBP. Благодаря высокой скорости расчета некоторых типов кодов LBP стал широко применяться для распознавания образов.

LBP представляет собой фильтр, обозначаемый как $LBP_{P,R}(x, y)$, который для каждой точки изображения рассчитывает код на основе значений точек в некоторой окрестности этой точки. В данном случае P – число точек, R – радиус окрестности.

Точки окрестности обозначим как g_i , где $i = \overline{0, P-1}$. При этом координаты точки рассчитываются как $(R \cdot \cos(\frac{i}{2\pi}); R \cdot \sin(\frac{i}{2\pi}))$. Обозначим изображение $f(x, y)$ или $f(g)$, если g это точка. Пусть

$$s_i(x, y) = \begin{cases} 1, & f(g_i) > f(g_c) \\ 0, & \text{иначе} \end{cases}$$

где g_c - точка с координатами (x, y) . Тогда

$$LBP_{P,R}(x, y) = \sum_{i=0}^{P-1} 2^i \cdot s_i(x, y).$$

Таким образом, последовательность $s_i(x, y)$, где $i = \overline{0, P-1}$, представляет собой двоичную последовательность кода LBP. Следовательно, $LBP_{P,R}(x, y) \in [0, 2^P - 1]$.

Для сравнения двух изображений в качестве векторов признаков используются гистограммы кодов LBP. В общем случае для каждого изображения строится гистограмма $H(l)$ для значений

$LBP_{P,R}(x, y)$, где $l = \overline{0, P-1}$. Существует несколько методов расчета расстояния между гистограммами. Например, расстояние Хи-квадрат:

$$\chi^2(H_1, H_2) = \sum_{i=0}^{B-1} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)},$$

где B - число кодов.

Существуют также модификации фильтра, описанные в работах [10, 13-15]:

Некоторые коды несут в себе больше информации, чем другие. Коды, в двоичной циклической записи которых число переходов между последовательностями «1» и «0» не превышает двух, обозначаются как «uniform», что соответствует слову «равномерный» [3]. Для заданного P существует $P \cdot (P-1) + 2$ равномерных значений. Модифицированный фильтр $LBP_{P,R}^{u2}$ в этом случае возвращает коды равномерных значений, добавляя только один код для неравномерных значений.

$$LBP_{P,R}^{u2}(x, y) = \begin{cases} \text{индекс кода, если он равномерный,} \\ P \cdot (P-1) + 2, & \text{иначе.} \end{cases}$$

4. Т.к. окрестность представляет собой круг, то можно найти группы кодов, инвариантных к повороту. Для каждого кода LBP существует P кодов, инвариантных к повороту, получаемых путем циклического сдвига P -битового числа. Для каждой такой группы в фильтр попадает минимальное значение кодов данной группы. Задача определения количества кодов, инвариантных к повороту является нетривиальной. Фильтр обозначается как $LBP_{P,R}^{\pi}$.

5. С учетом предыдущих двух свойств определяется также равномерный фильтр, инвариантный к повороту. Кодов LBP , обладающих одновременно двумя свойствами всего $P+2$, которые отличаются друг от друга числом бит, равных «1». В этом случае фильтр $LBP_{P,R}^{nu2}$ задается следующим образом:

$$LBP_{P,R}^{nu2}(x, y) = \begin{cases} \text{число единиц, если код равномерный} \\ P + 1, & \text{иначе} \end{cases}$$

В работе [14] был предложен метод сравнения гистограмм лиц. Изображение лица разбивается на $k \times k$ участков, для каждого из которых рассчитывается гистограмма. Итоговая гистограмма изображения лица определяется как конкатенация гистограмм участков изображения.

В работе [6] был предложен расширенный метод сравнения гистограмм лиц, основанный на взвешенной матрице. Задается матрица весов $k \times k$, каждый элемент которой соответствует участку изображения. Обозначим гистограмму j -

го участка изображения как H^j , $j = \overline{0, k^2 - 1}$.

Для каждого j -го участка задается вес w_j . Тогда можно определить модифицированное (взвешенное) расстояние Хи-квадрат следующим образом:

$$\chi_w^2(H_1, H_2) = \sum_{j=0}^{k^2-1} w_j \cdot \sum_{i=0}^{B-1} \frac{(H_1^j(i) - H_2^j(i))^2}{H_1^j(i) + H_2^j(i)},$$

где B - число кодов LBP .

В частности, в [6] предлагается использовать следующую матрицу для распознавания лиц (рис. 1).

2	1	1	1	1	1	2
2	4	4	3	4	4	2
1	1	1	2	1	1	1
0	1	1	2	1	1	0
0	1	1	1	1	1	0
0	1	1	2	1	1	0
0	1	1	1	1	1	0

Рис. 1 Матрица весов для распознавания лиц

2.5 Эксперименты по извлечению лиц

В процессе исследования коллекции было обнаружено около 4 тыс. лиц. Для оценки качества найденных объектов по их размерам было составлено общее распределение (рис. 2).

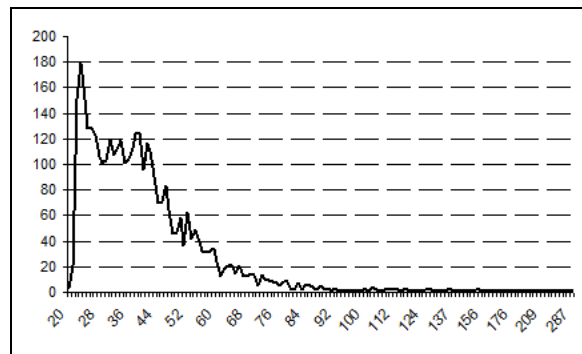


Рис. 2 Распределение размеров полной коллекции

Для оценки полноты и точности необходимо произвести экспертную оценку тестовой выборки (экспертная коллекция). В экспериментах использовалась выборка в 1070 изображений. Процесс оценивания был упрощен: вместо выделения всех лиц, на изображениях автоматически выделялись объекты полной коллекции, отсортированной вручную, после чего эксперту достаточно было отметить на изображении недостающие лица.

Ручная сортировка осуществлялась быстрым методом, позволяющим сортировать 100 объектов за 1 минуту. Кроме очевидных лиц и ложных объектов, найденных в «полной» коллекции, были определены дополнительные правила оценивания. К ложным объектам относились лица, слабо различимые (не видно глаз или рта), и лица людей, повернувшихся к камере на угол больше 90 градусов. К лицам относились различные изображения лиц: портреты, рисунки, бюсты.

Таблица 1 Результаты экспериментов по повышению точности и полноты

Название	Описание	Лица	Ошибки	Полнота	Точность	F-мера
ЭК	Экспертная коллекция	793	0	1,000	1,000	1,000
ПК	Полная коллекция	540	973	0,681	0,357	0,468
ТК	Точная коллекция	439	289	0,554	0,603	0,577
LBP_24_1	$LBP_{24,3}^{riu2}, E_1$	529	484	0,667	0,522	0,586
LBP_24_2	$LBP_{24,3}^{riu2}, E_2$	492	294	0,620	0,626	0,623
LBP_24_W_Q_1	$LBP_{24,3}^{riu2}$, веса, сжатие, E_1	522	364	0,658	0,589	0,622
LBP_24_W_Q_2	$LBP_{24,3}^{riu2}$, веса, сжатие, E_2	499	319	0,629	0,610	0,619
LBP_24_W_1	$LBP_{24,3}^{riu2}$, веса, E_1	506	292	0,638	0,634	0,636
LBP_24_W_2	$LBP_{24,3}^{riu2}$, веса, E_2	483	226	0,609	0,681	0,643
LBP_16_1	$LBP_{16,3}^{riu2}, E_1$	532	492	0,671	0,520	0,586
LBP_16_2	$LBP_{16,3}^{riu2}, E_2$	512	332	0,646	0,607	0,626
LBP_16_W_Q_1	$LBP_{16,3}^{riu2}$, веса, сжатие, E_1	520	370	0,656	0,584	0,618
LBP_16_W_Q_2	$LBP_{16,3}^{riu2}$, веса, сжатие, E_2	499	306	0,629	0,620	0,625
LBP_16_W_1	$LBP_{16,3}^{riu2}$, веса, E_1	510	305	0,643	0,626	0,634
LBP_16_W_2	$LBP_{16,3}^{riu2}$, веса, E_2	488	230	0,615	0,680	0,646

В процессе классификации объекты преобразовывались к размеру 64×64 . Эксперименты проводились и использованием нескольких фильтров с различными параметрами расчета гистограммы и с различными обучающими множествами. Для вычисления векторов признаков использовались фильтры $LBP_{16,3}^{riu2}$ и $LBP_{24,3}^{riu2}$. В одних случаях применялась

матрица весов (рис. 1), в других случаях вектор признаков сжимался до размера 200.

В качестве множества лиц использовался набор из 18 изображений лиц. Кроме того, было задано 2 множества ложных объектов (8 и 26 изображений, E_1 и E_2 соответственно).

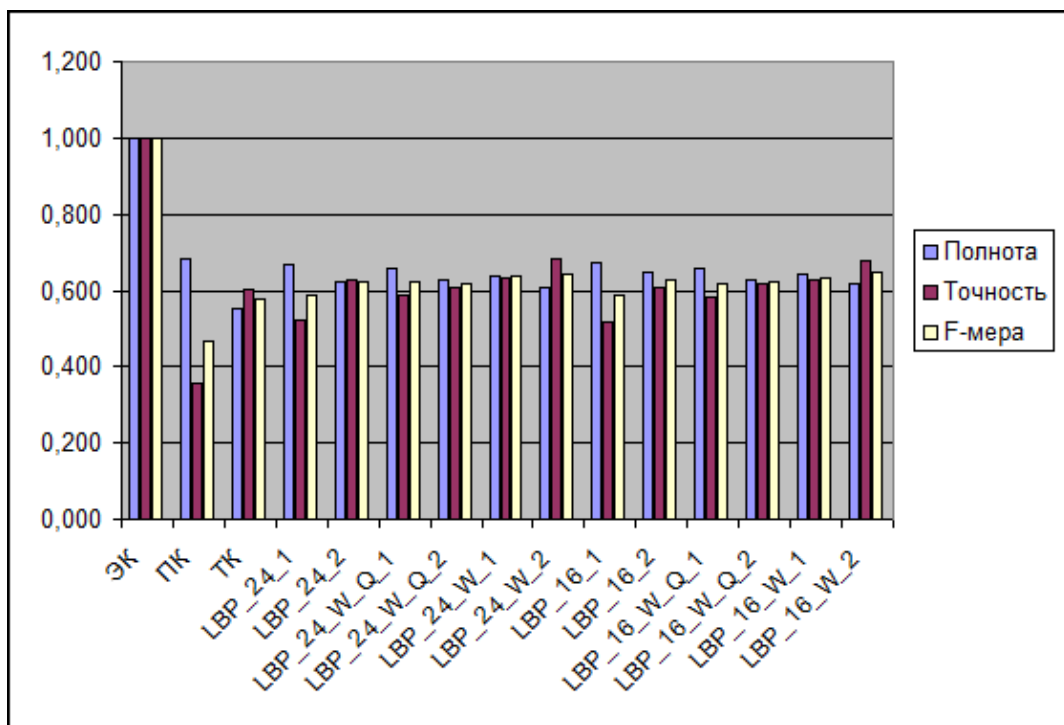


Рис. 3 Диаграмма сравнения полноты и точности обнаружения лиц

Результатом каждого эксперимента являлась отдельная коллекция объектов. Полученные коллекции оценивались полнотой и точностью, для расчета которых объекты коллекций сопоставлялись с объектами экспертной коллекции. Результаты экспериментов, позволяющие сравнить качество полученных коллекции с соответствующими параметрами «полной» и «точной» коллекций, представлены в таблице 1 и на рисунке 3. Описание каждого эксперимента содержит параметры фильтра LBP, использованные методы расчета гистограмм, а также множество ложных объектов.

Проведенные эксперименты показали, что гистограммы, полученные с помощью фильтров $LBP_{16,3}^{nu2}$ и $LBP_{24,3}^{nu2}$, матрицы весов и расширенной обучающей выборки обладают наилучшей селективной способностью с точки зрения F-меры [1]. Кроме того, в большинстве случаев, расширение обучающей выборки позволило сократить количество ошибок и, следовательно, точность и F-меру отсортированной коллекции.

2.6 Эксперименты по распознаванию лиц

Таблица 2
Тестовое множество изображений для распознавания лиц



Современные исследования демонстрируют высокие показатели точности распознавания лиц

размерами выше 100. В данном случае требуется отдельный подход.

Для экспериментов использовалась следующая коллекция изображений (таблица 2), обозначим данный набор как T .

При этом были обнаружены следующие пары лиц, соответствующих одному и тому же человеку: $\{3, 14\}$, $\{4, 13\}$ и $\{7, 9\}$. Обозначим данное множество как T^+ . Кроме того, очевидна пара ложных объектов, которые должны быть близки друг к другу: $\{1, 15\}$ (T^-). Дополним множества T^+ и T^- зеркально отраженными элементами данных множеств.

Для исследования работы различных фильтров LBP используется следующий алгоритм:

1. Для каждого объекта $t_i \in T$ находится ближайший к нему объект $t_i^* \in T$. Т.е.

$$t_i^* = \arg \min_{t_j} \{\rho_f(t_i, t_j) | t_j \in T/t_i\}$$

2. Полученное таким образом множество пар $T^P = \{<t_i, t_i^* > | t_i \in T\}$ сопоставляется с известными парами лиц и ложных объектов.

3. Обозначим

$$h(t_i, t_j) = \begin{cases} 1, <t_i, t_j > \in T^+ \text{ или } <t_i, t_j > \in T^- \\ 0, \text{ иначе} \end{cases}$$

4. Будем рассчитывать точность следующим

$$\text{образом: } Re = \frac{\sum_{t_i \in T} h(t_i, t_i^*)}{|T^+| + |T^-|}$$

Таблица 3
Результаты экспериментов по распознаванию лиц

Обозначение	Точность Re
$LBP_{8,1}$	3/8
$LBP_{16,1}$	2/8
$LBP_{8,2}$	4/8
$LBP_{16,2}$	4/8
$LBP_{8,3}$	4/8
$LBP_{16,3}$	6/8
Взвешенный $LBP_{16,3}^n$	4/8
Взвешенный $LBP_{16,3}^{nu}$	3/8
Взвешенный $LBP_{16,3}^u$	5/8
Взвешенный $LBP_{16,3}$	8/8

В процессе исследования были рассмотрены различные варианты фильтров LBP. В таблице 3

приведены показатели точности использованных фильтров относительно заданного множества T .

Таким образом, наибольшая точность достигается для взвешенного фильтра $LBP_{16,3}$.

3 Аннотирование на основе текстур

3.1 Общая схема аннотирования

В данной работе рассматривается процесс аннотирования за счет текстур, который в общем случае включает в себя следующие шаги:

1. Задается набор текстур, каждая из которых принадлежит одному из нескольких заданных классов.
2. Каждому классу присваивается набор меток.
3. Для каждого изображения выполняется поиск текстур из заданного набора.
4. Для каждой найденной текстуры к изображению добавляется набор меток соответствующего класса.

Для выполнения текстурного поиска требуется классификатор. В данной работе предлагается классификатор на основе метода моментов.

3.2 Метод моментов

В работе [17] был предложен метод сегментирования текстур на основе моментов и описан соответствующий математический аппарат. Пусть $f(x, y)$ представляет собой изображение длиной W и шириной H . Область значений $f(x, y)$ - $[0; 1]$, что соответствует области $[0; 255]$ обычного монохромного изображения. Метод моментов заключается в том, что для каждой точки изображения $f(x, y)$ рассчитывается набор моментов и производных от них характеристик в пределах некоторого окна с центром в данной точке. Таким образом, формируется набор изображений, соответствующих набору признаков.

Момент $M_{p,q}$ с размером окна W_M рассчитывается следующим образом:

$$M_{p,q}(i, j) = \sum_{(a,b) \in W_{i,j}^M} f(a, b) \cdot x_a^p \cdot y_b^q \quad (1),$$

$$\text{где } x_a = \frac{a-i}{W_M/2}; y_b = \frac{b-i}{W_M/2} \quad (2),$$

где $W_{i,j}^M$ - окно размером W_M с центром в точке (i, j) .

В той же работе [17] утверждалось, что набор моментов в «чистом» виде не годится для сегментирования, поэтому автором был предложен улучшенный набор признаков:

$$F_{p,q}(i, j) = \frac{1}{W_F^2} \sum_{(a,b) \in W_{i,j}^F} |\tanh(\sigma(M_{p,q}(a, b) - \overline{M_{p,q}}))| \quad (3),$$

$$\overline{M_{p,q}}(i, j) = \frac{1}{W_F^2} \sum_{(a,b) \in W_{i,j}^F} M_{p,q}(a, b) \quad (4),$$

где $W_{i,j}^F$ - окно размером W_F с центром в точке (i, j) .

Степени моментов p и q задаются таким образом, чтобы их сумма не превышала некоторого значения O , которое представляет собой порядок моментов. Число признаков, таким образом, зависит от O . В данной работе рассматривались моменты порядка не выше 2-го.

Другими параметрами для расчета характеристик являются размеры окон W_M и W_F и коэффициент σ . Авторами данного метода были предложены значения данных параметров: 9, 49 и 0.01 соответственно. К сожалению, данные значения были обоснованы методом «проб и ошибок» без какого-либо аналитического или практического обоснования. Поэтому в данной работе был проведен анализ влияния значений данных параметров на результат сегментации.

Для экспериментов использовался алгоритм сегментации, производный от алгоритма, предложенного в [17].

1. Тестовое изображение «склеивается» из нескольких текстур (пример на рисунке 4). Каждая текстура представлена квадратным участком. Площади участков текстур одинаковы.
2. Рассчитываются характеристики, представляющие собой N изображений, где N – число характеристик.
3. Случайно отбираются 6 % точек, равномерно распределенных по изображению.
4. Для выбранных точек производится кластеризация методом k -средних. Число кластеров соответствует числу текстур. Авторы метода предложили алгоритм кластеризации CLUSTER [12].
5. Полученные центры кластеров как результат работы алгоритма k -средних используются для сегментации всего изображения. Каждому пикселю ставится в соответствие номер того кластера, к которому ближе всего располагается соответствующий вектор признаков.
6. Определим точность сегментации ρ как доля правильно сегментированных пикселей. Поскольку алгоритм k -средних произвольно определяет начальные центры кластеров, к алгоритму добавляется промежуточный шаг, на котором определяется соответствие номеру текстуры и номеру кластера. Будем считать, что кластер с номером i будет соответствовать текстуре j , если большая часть пикселей класса i будет располагаться внутри участка текстуры j .

Пример сегментации изображения (рис. 4), «склеенного» из 2-х текстур, представлен на рисунке 5. Точность сегментации составила 95,3%.

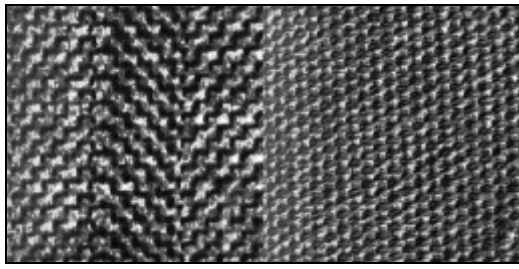


Рис. 4 Тестовое изображение, «склеенное» из 2-х текстур альбома Бродаца [7].



Рис. 5. Пример бинарной сегментации изображения на рисунке 4.

Анализ результатов сегментации заключается в сравнении значений точности для определенного набора параметров $\{W_M, W_F, \sigma\}$. В данной работе были выбраны следующие диапазоны значений параметров: [9; 49] с шагом 5 для W_M и W_F , [0.005; 0.02] с шагом 0.005. Анализ осуществлялся над изображением на рисунке 4.

В таблице 4 представлены результаты анализа, отсортированные по убыванию точности сегментации.

Таблица 4 Результаты анализа влияния параметров на точность сегментации

W_M	W_F	σ	ρ (%)
9	49	0,01	95,285
9	39	0,005	95,1782
9	39	0,02	95,1752
9	44	0,005	95,1355
9	49	0,015	95,1324
9	49	0,005	95,1141
9	44	0,01	95,0897
9	39	0,015	95,0745
9	44	0,02	95,0714
9	39	0,01	95,0592

Таким образом, экспериментально показана правильность предложенных авторами значений параметров.

3.3 Технические аспекты реализации метода моментов

С точки зрения теории поиск текстур требует вычисления значений характеристик для каждого

пикселя искомого изображения. Реализация формул (1), (3), (4) прямым способом включает в себя 4 вложенных цикла. Следовательно, общая сложность вычислений равна $O(N^2M^2)$, где N – линейный размер изображения, M – размер окна для расчета характеристик. Проведенные эксперименты показали, что при $N=128$ и $M=49$ расчет характеристик занимает в среднем 5-10 минут на виртуальной машине при реализации на C++. Поэтому в данной работе метод расчета моментных характеристик был изменен.

Формула (1) представляет собой свертку с ядром размера $W_M \times W_M$, при этом значения ядра рассчитываются по формуле (2). По аналогии со сверткой 2-х непрерывных функций можно вычислить свертку через преобразования Фурье. Такой метод, в частности, реализован в функции filter2D библиотеки openCV [11]. Для размера окна больше 13x13 сложность данной функции равна $O(N^2 \log N)$ и не зависит от размера окна. Для меньших размеров свертка рассчитывается прямым способом, однако время вычислений по-прежнему остается небольшим. Обозначим ядро для формулы (1) как C_M . В общем случае ядро также представляет собой изображение.

Формулы (3) и (4) включают в себя свертку с один и тем же ядром C_F размера $W_F \times W_F$. Данное ядро вычисляется следующим образом:

$$C_F(x, y) = \frac{1}{W_F^2}.$$

Кроме того, в формуле (1) используется преобразование с помощью логистической функции, которое можно выделить как отдельную функцию над изображением: $T(f(x, y)) = |\tanh(\sigma(f(x, y)))|$.

По аналогии со сверткой непрерывных функций обозначим оператор свертки изображений следующим образом: $(f * g)(x, y)$. Тогда $f * g = \text{filter } 2D(f, g)$, где f представляет собой исходное изображение, а g – ядро фильтра. С учетом вышеперечисленных обозначений можно переписать формулы для расчета моментных характеристик следующим образом:

$$F_{p,q} = T(f * C_M - (f * C_M) * C_F) * C_F$$

Сложность преобразования $T(f(x, y))$ составляет $O(N^2)$, поэтому общая сложность вычислений составляет $O(N^2 \log N)$. Для сравнения, на той же виртуальной машине, время с теми же начальными условиями время вычисления составляет меньше секунды.

В экспериментах по поиску текстур на изображении с помощью классификатора использовались изображения размером 1216*960 пикселей. Среднее время расчета характеристик при таких исходных данных составило 1 минуту.

3.4 Классификатор на основе метода моментов

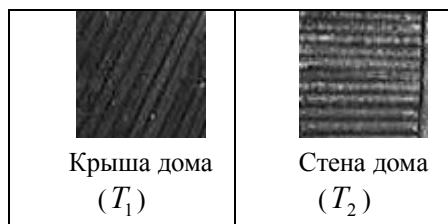
Для поиска текстур предлагается следующий классификатор:

1. Для каждой текстуры в обучающей выборке рассчитывается центральный вектор (центр текстуры) как средний вектор среди векторов признаков всех пикселей данной текстуры.
2. Для каждой текстуры задается параметр ε , соответствующий максимально допустимому расстоянию от векторов признаков найденных пикселей до центра соответствующей текстуры. Параметры ε для каждой текстуры подбираются так, чтобы доля правильно классифицированных пикселей была наибольшей, а доля неправильно классифицированных пикселей наименьшей.
3. Для каждого изображения вычисляются карты признаков (по числу компонент вектора). Соответствующие точки карт признаков образуют вектор признаков.
4. Для каждого изображения вычисляется карта текстур. Размер карты соответствует размеру изображения. Для каждой точки изображения определяется, к какой текстуре принадлежит соответствующий вектор признаков. Если таких текстур несколько, данная точка карты принимает в качестве значения номер ближайшей текстуры. Если вектор признаков не соответствует ни одной из текстур, точка карты принимает значение -1.
5. Точки полученной карты текстур объединяются в сегменты, которые состоят из соседних точек с одинаковым номером текстуры. Точки со значением -1 в сегменты не объединяются.
6. Для каждого сегмента определяется его размер. Если размер сегмента равен или превышает некоторое пороговое значение, считается, что соответствующая текстура найдена на текущем изображении. В данной работе использовалось пороговое значение размера сегмента 100.

3.5 Эксперименты по поиску текстур

Для эксперимента было отобрано 100 изображений из коллекции фотографий строительства Беломорско-Балтийского канала. Для поиска использовались следующие текстуры:

Таблица 5
Изображения, использованные в эксперименте по текстурному поиску



Эксперимент заключается в выявлении вхождения каждой текстуры T_j в каждое изображение I_i .

С помощью эксперта вручную отмечаются вхождения текстур $E_{ij} = \begin{cases} 1, T_j \text{ найдено в } I_i \\ 0, \text{ иначе} \end{cases}$. С другой

стороны с помощью классификатора автоматически определяются вхождения каждой текстуры F_{ij} (аналогично E_{ij}). После чего определяется флаг релевантности текстуры R_{ij} (1-текстура T_j найдена на изображении I_i и при этом обнаружена экспертом), который рассчитывается как $E_{ij} \cdot F_{ij}$. Тогда полнота и точность поиска текстуры T_j рассчитывается следующим образом:

$$\text{Pr}_j = \frac{\sum_i R_{ij}}{\sum_i F_{ij}}; \text{Re}_j = \frac{\sum_i R_{ij}}{\sum_i E_{ij}}$$

Аналогично можно рассчитать общую полноту и точность поиска:

$$\text{Pr} = \frac{\sum_{i,j} R_{ij}}{\sum_{i,j} F_{ij}}; \text{Re} = \frac{\sum_{i,j} R_{ij}}{\sum_{i,j} E_{ij}}$$

На рисунке 6 представлены основные результаты эксперимента.

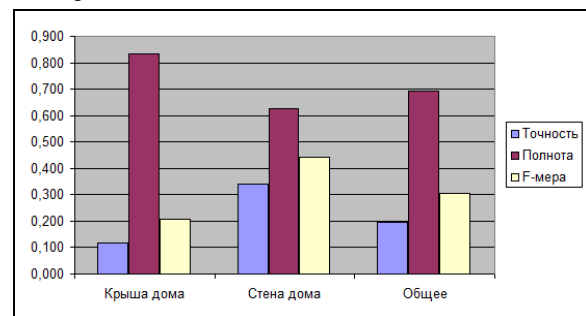


Рис. 6 Результаты поиска текстур

4 Заключение

В результате работы был предложен улучшенный метод обнаружения лиц на основе алгоритма Виолы-Джонса, модификации алгоритма ЛБШ и обучающей выборки. Несмотря на то, что в экспериментах использовались относительно небольшие обучающие выборки, метод показал более высокие характеристики по сравнению с алгоритмом Виолы-Джонса. Кроме того, данный метод ориентирован на объекты низкого разрешения. В дальнейшем планируется исследовать метод на коллекциях более высокого разрешения.

Предложен автоматизированный метод аннотирования изображений за счет лиц с помощью эксперта на основе модификации алгоритма ЛБШ.

Предложен классификатор и метод аннотирования изображений за счет текстур с помощью

метода моментов. В дальнейшем планируется модифицировать классификатор с целью повышения качества текстурного поиска.

На основе результатов проведенных экспериментов был построен прототип поисковой системы по поиску фотографий содержащих заданные объекты в электронной коллекции исторических фотографий со строительства Беломорско-Балтийского канала.

Литература

- [1] Агеев. М, Кураленок И. Официальные метрики РОМИП'2004 [Электронный ресурс]. URL: http://romip.ru/docs/romip_metrics.pdf (дата обращения: 10.08.2012).
- [2] Маслий Р. В. Использование локальных бинарных шаблонов для распознавания лиц на полутоновых изображениях // Наукові праці ВНТУ, 2008, № 4. – Винницкий национальный технический университет.
- [3] Петрук В. И., Самородов А. В., Спиридонов И. Н. Применение локальных бинарных шаблонов к решению задачи распознавания лиц // Вестник МГТУ им. Н. Э. Баумана. Сер. «Приборостроение», 2011.
- [4] Пименов В. Ю. Простые методы поиска изображений по содержанию / Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2010. – Казань: Казан. ун-т, 2010. – 210. с., с 69 – 79.
- [5] Слесарев А. В., Мучник И. Б., Михалев Д. К. Яндекс на РОМИП 2010: Поиск похожих изображений и дубликатов / Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2010. – Казань: Казан. ун-т, 2010. – 210. с., с 148 – 153.
- [6] Ahonen T., Hadid A., Pietikäinen M. Face Recognition with Local Binary Patterns // Machine Vision Group, Infotech Oulu.
- [7] Brodatz P. Textures: a Photographic Album for Artists and Designers. – New York: Dover Publications, 1966.
- [8] Cascade Classification — opencv v2.1 documentation [Электронный ресурс]. URL: http://opencv.willowgarage.com/documentation/cpp/cascade_classification.html#cv-cascadeclassifier-detectmultiscale (дата обращения: 15.05.2011).
- [9] Degtyarev N., Seredin O. Comparative Testing of Face Detection Algorithms. In: Elmoataz A., Lezoray O., Nouboud F., Mammazz D., Meunier J. (eds.) ICISP 2010. LNCS, vol. 6134, pp. 200 – 209, Springer, Heidelberg (2010).
- [10] Guo Z., Zhang L., Zhang D. Rotation invariant texture classification using LBP variance (LBPV) with global matching // Pattern Recognition 43 (2010) pp. 706 –719.
- [11] Image Filtering — opencv v2.1 documentation [Электронный ресурс]. URL: http://opencv.willowgarage.com/documentation/cpp/image_filtering.html#cv-filter2d (дата обращения: 15.05.2011).
- [12] Jain A. K., Dubes R. C. Algorithms for Clustering Data. – Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- [13] Mäenpää T. The local binary pattern approach to texture analysis – extensions and applications. – Infotech Oulu and Department of Electrical and Information Engineering, University of Oulu, 2003.
- [14] Ojala T., Pietikäinen M., Harwood D. A Comparative Study of Texture Measures with Classification Based on Feature Distributions // Pattern Recognition, vol. 29, pp. 51-59. – 1996.
- [15] Ojala T., Pietikäinen M., Mäenpää T. A Generalized Local Binary Pattern Operator for Multiresolution Gray Scale and Rotation Invariant Texture Classification // Machine Vision and Media Processing Unit Infotech Oulu. – University of Oulu.
- [16] Picasa [Электронный ресурс]. URL: <http://picasa.google.com> (дата обращения: 15.06.2012).
- [17] Tuceryan M. Moment Based Texture Segmentation // Pattern Recognition Letters, vol. 15, pp. 659-668, July 1994.
- [18] Viola P., Jones M. Robust Real-time Object Detection // Second International Workshop on Statistical and Computational Theories of Vision – Modelling, Learning, Computing and Sampling. – Vancouver, Canada, 2001.
- [19] Wechsler H. Reliable face recognition methods: system design, implementation and evaluation, Springer, 329 p. (2007).

Tagging of digital historical images

A. N. Talbonen, A. A. Rogov

This article considers the methods of digital historical image annotation due to the presence of people with expert help and textures by using the method of moments. An improved method of face detection based on the combination of Viola-Jones algorithm and local binary patterns is proposed.