

# Методы распознавания лиц

Ю. Лифшиц\*

4 декабря 2005 г.

## Содержание

<b>1</b>	<b>Применение алгоритмов распознавания лиц</b>	<b>2</b>
<b>2</b>	<b>Особенности распознавания лиц</b>	<b>2</b>
2.1	Специфика задачи . . . . .	2
2.2	Абстрактная постановка задачи . . . . .	3
2.3	Этапы алгоритма распознавания лиц . . . . .	3
<b>3</b>	<b>Метод сопоставления графов</b>	<b>4</b>
3.1	40 фильтров Габора . . . . .	4
3.2	Создание “единого портрета” . . . . .	4
3.3	Нахождение контрольных точек . . . . .	5
3.4	Классификация . . . . .	5
3.5	Преимущества . . . . .	5
<b>4</b>	<b>Собственные лица</b>	<b>6</b>
4.1	Понижение размерности . . . . .	6
4.2	Построение базиса пространства лиц . . . . .	6
4.3	Классификация . . . . .	6

---

\*Законспектировал Д. Паращенко

# 1 Применение алгоритмов распознавания лиц

На первый взгляд кажется, что распознавание лиц применимо лишь в следующих областях:

- Охранные системы
- Криминалистика
- Компьютерная графика

На самом деле, спектр применения этих алгоритмов намного шире:

- Взаимодействие компьютер-человек
- Виртуальная реальность, компьютерные игры
- Водительские права, паспорт
- Контроль над иммиграцией
- Персонализация бытовых устройств
- Шифрование данных
- Электронная коммерция
- Криминалистика

Все вышесказанное делает актуальной задачу распознавания лиц.

## 2 Особенности распознавания лиц

### 2.1 Специфика задачи

При решении задачи распознавания лиц возникают две проблемы.

Во-первых, любая картинка представляет собой массив пикселей. В то же время один пиксель картинки ничего не значит (его цвет можно изменить, и никто не заметит разницы). Это делает такое представление картинок избыточным и неэкономичным. Таким образом, для эффективного распознавания лиц необходимо разработать некоторый компактный и удобный формат представления картинок. На сегодняшний день известно множество способов сжатия изображений с потерями, но используемый в нем формат не удобен для классификации фотографий людей, хотя бы, потому что для решения задачи распознавания лиц требуется, опять-таки, гораздо меньше информации. Это связано в первую очередь с тем, что нет необходимости определять, как выглядит данный человек из коллекции, а требуется решить обратную задачу: какой человек из коллекции выглядит данным образом.

Вторая проблема заключается в том, что одно и то же лицо может быть сфотографировано при различных внешних факторах, таких как свет, поза, эмоции.

## 2.2 Абстрактная постановка задачи

В дальнейшем будем считать, что все фотографии имеют размер  $100 \times 100$  пикселей с 256 оттенками серого.

Можно придумать много вариантов постановки задачи распознавания лиц. Один из таких вариантов представлен ниже.

Предположим, что имеется некоторая тренировочная коллекция, состоящая из 400 фотографий (по 10 фотографий для 40 людей при разных условиях).

Тогда задачу распознавания лиц можно сформулировать следующим образом. Имеется некоторое новое изображение в оговоренном выше формате. Необходимо выдать один из следующих ответов:

- Изображение не является лицом
- Изображение является лицом такого-то человека из коллекции
- Изображение является лицом, но его нет в коллекции. В этом случае его можно добавить в коллекцию

## 2.3 Этапы алгоритма распознавания лиц

На первом этапе алгоритм выделяет из общего изображения лицо. Затем происходит нормализация изображения. К нормализации изображения относятся следующие действия:

- Изменение разрешения изображения до  $100 \times 100$
- Преобразование цветов к 256 оттенкам серого
- Изменение суммарной яркости изображения до некоторого среднего значения

Для некоторых алгоритмов требуется, чтобы лицо на картинке располагалось как можно более вертикально. В таком случае поворот лица на нужный угол происходит именно при нормализации.

Следующим этапом алгоритма распознавания является выделение характеристик имеющегося лица. Выделяемые характеристики сильно зависят от алгоритма распознавания, поэтому их примеры будут приведены позже. Заметим, что после выделения характеристик картинка больше не нужна.

Последним этапом распознавания является применение классификатора, который по имеющимся характеристикам выдает ответ на задачу.

## 3 Метод сопоставления графов

### 3.1 40 фильтров Габора

Сверткой функции  $f(x)$  называется функция  $S(y) = \int G(x - y)f(x)d^2x$ . При этом,  $G(x)$  называется ядром свертки. Дискретным аналогом операции свертки является взвешенная сумма.

Рассмотрим вектора  $k_i = (k_\nu \cos \phi_\mu, k_\nu \sin \phi_\mu)$ , где  $k_\nu = 2^{-\frac{\nu+2}{2}}$  и  $\phi_\mu = \mu \frac{\pi}{8}$  ( $0 \leq \nu \leq 4, 0 \leq \mu \leq 7$ ). Нетрудно видеть, что 40 векторов  $k_i$  различаются по длинам (5 значений  $\nu$ ) и по направлениям (8 значений  $\mu$ ).

Теперь перейдем к фильтрам Габора. Фильтрами Габора называются свертки со следующими ядрами:

$$G_i = \frac{k_i^2}{\sigma^2} \exp(-\frac{k_i^2 x^2}{2\sigma^2}) [\exp(ik_i x) - \exp(\frac{\sigma^2}{2})]$$

Итого, мы имеем 40 ядер Габора.

Фильтры Габора обладают прекрасной геометрической устойчивостью, а именно, они устойчивы к операциям масштабирования, поворота, изменения яркости или контрастности.

### 3.2 Создание “единого портрета”

У всех людей отношения расстояния между глазами к высоте носа примерно одинаковы. Поэтому, целесообразно выделить на лице некоторые контрольные точки, такие как нос, глаза, брови, рот, скулы, щеки, подбородок. После этого для идентификации того или иного человека, достаточно считать значения фильтров Габора именно в контрольных точках.

Выделим на произвольном лице некоторые контрольные точки. Для того, чтобы определить, где у заданного человека находятся эти точки, необходимо иметь некоторые представления о их расположении. Для этого построим “единый портрет”.

Зафиксируем набор интересующих нас контрольных точек (именно этот этап определяет качество алгоритма распознавания). Затем для каждого изображения из коллекции укажем, где находится каждая из контрольных точек, и посчитаем значения 40 фильтров Габора в этих точках. Вектор, состоящий из 40 значений фильтров Габора, посчитанных в конкретной точке называется *jet*-ом этой точки. Теперь усредним расстояния между контрольными точками. Также усредним *jet*-ы.

Итого, мы получили некоторый граф, в котором вершинам соответствуют контрольные точки, а длины ребер равны средним расстояниям между данными контрольными точками. Кроме того, в каждой вершине хранится один “средний” *jet*.

Полученный таким образом граф называется “единым портретом”.

### 3.3 Нахождение контрольных точек

После получения на вход нового лица прежде всего надо найти положение контрольных точек на нем.

Стоит пояснить, в каком смысле мы ищем положение контрольных точек. Для каждой контрольной точки известен ее  $jet$ . Мы хотим найти такой вектор точек, чтобы  $jet$  каждой точки вектора был как можно ближе к  $jet$ 'у соответствующей контрольной точки. При этом, мы также хотим, чтобы расстояния между выбранными точками были как можно более пропорциональны длинам ребер графа "единого портрета". Таким образом, можно ввести некоторую вектора от набора точек, и затем ее минимизировать (найти вектор, на котором достигается минимальное значение).

Имеется довольно много методов минимизации различных функций. Для данного случая подходит очень простой метод. Сначала большими шагами параллельно перемещаем решетку "единого портрета" и сравниваем получающиеся  $jet$ 'ы контрольных точек с эталонными. После того, как мы выяснили примерное расположение решетки, поворачиваем и порастягиваем ее. Затем начнем независимо перемещать каждую контрольную точку на незначительное расстояние. Этот метод находит не полностью минимизирующий набор точек, но в задаче распознавания лиц эти точки можно считать контрольными точками изображения.

### 3.4 Классификация

После того, как мы нашли контрольные точки изображения, мы имеем вектор нового изображения, являющийся набором  $jet$ 'ов всех контрольных точек. Требуется определить, какому лицу (человеку) соответствует данный вектор. Для этого применяются стандартные методы классификации, например, метод  $k$  соседей.

### 3.5 Преимущества

Метод классификации графов обладает рядом преимуществ:

- Прежде всего, он легко обучаем, т.е. для достижения хороших результатов ему необходима маленькая исходная коллекция
- Во-вторых, он обладает хорошей устойчивостью к позам

## 4 Собственные лица

Рассмотрим принципиально иной подход, основанный на идее извлечения не локальных, а глобальных характеристик.

### 4.1 Понижение размерности

Прежде всего заметим, что картинки имеют размерность 10000. Портретов намного меньше, чем картинок.

Сделаем предположение, что все портреты располагаются в некоторой плоскости, например 100-мерной.

В таком случае лицо можно разложить на две составляющие:

1. Проекция на 100-мерную плоскость
2. Расстояние до этой плоскости (шум)

Таким образом, мы будем хранить координаты портрета в выбранной плоскости, а также базисные вектора этой плоскости в исходном пространстве размерности 10000.

Описанный выше подход обусловлен рядом причин:

- Уменьшаются вычисления и требования к памяти
- Можно обойтись меньшей тренировочной коллекцией
- Отбрасываются незначительные детали

### 4.2 Построение базиса пространства лиц

Мы имеем  $n$  портретов и соответствующие им вектора  $x_i$  из пространства изображений ( $ImageSpace, |ImageSpace| = 10000$ ).

Будем строить базис подпространства лиц ( $FaceSpace, |FaceSpace| = 100$ ) постепенно. Первый вектор базиса  $y_1$  выберем так, чтобы он как можно сильнее “различал” имеющиеся лица. Для этого надо максимизировать дисперсию  $\sum_{i=1}^n \frac{x_i - y_1}{n}$ . Затем перейдем к ортогональному дополнению  $y_1$  в  $ImageSpace$  и повторим действия.

Таким образом мы построим базис  $y_i, 1 \leq i \leq 100$ .

### 4.3 Классификация

Нам предоставляют вектор  $z$  пространства  $ImageSpace$ . Найдём его разложение по базису пространства  $FaceSpace$ . Теперь нам требуется определить, какому лицу из имеющейся коллекции соответствует этот вектор. Для этого, опять-таки, можно воспользоваться методом  $k$  соседей.