

Сравнительный анализ алгоритмов распознавания лиц в задаче визуальной идентификации

© О.В. Рогозин, С.А. Кладов

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

Статья посвящена двум подходам к распознаванию лиц, реализованным в программном комплексе визуальной идентификации. Описаны соответствующие алгоритмы, приведены блок-схемы и фрагменты их программной реализации, проанализированы результаты применимости на основе проведенных исследований.

Ключевые слова: *распознавание лиц, алгоритм EigenFace, фильтр Габора, визуальная идентификация.*

Система идентификации по голосовым и визуальным данным предназначена для распознавания личности в местах массового скопления людей, в зонах контроля, на стратегических объектах. Функциональные возможности системы позволяют повысить надежность идентификации за счет ее многоуровневой структуры, основанной на использовании различных биометрических характеристик. Важной частью этой структуры является блок распознавания лиц, в котором реализованы два независимых алгоритма распознавания, отличающиеся точностью, скоростью и ресурсоемкостью. Различия в характеристиках алгоритмов позволяют оценивать целесообразность применения их в конкретной ситуации, в то время как одновременное использование увеличит надежность идентификации. Далее рассмотрим сами алгоритмы, некоторые аспекты их реализации и интегрирования в систему. Фрагменты программного кода написаны на языке C#.

Алгоритмы анализа изображений. Основная идея алгоритма EigenFace заключается в нахождении «среднего лица», т. е. обобщенного и усредненного варианта всех фотографий пользователей в базе данных. С помощью полученного «среднего лица» для каждой фотографии пользователя находится «разностное лицо», т. е. разница между ней и «средним лицом». Полученное «разностное лицо» представляет собой те черты лица, которые реже всего встречаются на остальных изображениях в базе данных. При поступлении изображения на вход подсистемы для него вычисляется «разностное лицо» и сравнивается с каждым «разностным лицом» в базе данных с помощью евклидова расстояния:

$$\rho(x, y) = \|x - y\| = \sqrt{\sum_{k=1}^n (x_k - y_k)^2},$$

где x, y — векторы размерности n .

Последовательность обработки изображений с помощью алгоритма EigenFace.

1-й шаг: получение массива фотографий, которое происходит с помощью признаков Хаара и библиотеки OpenCV. Изображение хранится в базе данных в виде матрицы $I[m, n]$ значений цвета его пикселей.

```
I = _img.HaarDetectObjects(_Cascade, _Storage).
```

2-й шаг: Получение матрицы всех изображений. Каждая фотография преобразовывается в вектор построчно:

$$ImageVector[i * m + j] = Incoming[i, j],$$

где *Incoming* — исходное изображение в матричном виде, а *ImageVector* — в векторном. Затем полученные векторы изображений записываются в общую матрицу *Common* по столбцам.

3-й шаг: вычисление среднего лица для базы данных изображений:

$$Avg[i] = \frac{1}{N} \sum_{j=0}^{n-1} Common[i, j],$$

где *Avg* — вектор среднего лица, N — общее количество изображений в базе данных.

```
// A - изначально заполнена 0
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
    {
        A[i, j] += I[i, j];
    }
}
for (int i = 0; i < iLength; i++)
{
    A[i, j] /= N;
}
```

4-й шаг: вычисление матрицы разностных изображений в соответствии с выражением $Dif[i, j] = Common[i, j] - Avg[i]$

```
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
    {
        Dif[i, j] = I[i, j] - A[i, j];
    }
}
```

}
}

5-й шаг: вычисление ковариационной матрицы Cov и ее собственных векторов v :

$$Cov = Dif^T * Dif .$$

6-й шаг: вычисление собственных векторов u матрицы разностных изображений Dif и вектора их весов w :

$$u = Dif * v,$$

$$w = u * Dif.$$

Вектор w весов собственных векторов каждого изображения хранится в базе данных. Далее на их основе проводится распознавание. Каждое входное изображение преобразуется по аналогичному алгоритму. Распознавание заключается в вычислении евклидова расстояния для каждой пары $(wBase, wInc)$ и выборе среди них минимального. Здесь $wBase_i$ — вектор весов собственных векторов i -го изображения из базы данных, а $wInc$ — входного изображения.

$$RecogIndex = \min(\rho(wBase_i, wInc)), i \in [1; N].$$

Алгоритм распознавания на основе фильтров Габора. Фильтр Габора (рис. 1) — линейный электронный фильтр, импульсная переходная характеристика которого определяется в виде гармонической функции, помноженной на гауссиан:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right),$$

где $x' = x \cos \theta + y \sin \theta$; $y' = -x \sin \theta + y \cos \theta$.

В этом уравнении λ представляет собой длину волны множителя косинуса, θ определяет ориентацию нормали параллельных полос функции Габора в градусах, ψ — сдвиг фаз в градусах и γ — коэффициент сжатия, характеризующий эллиптичность функции Габора.

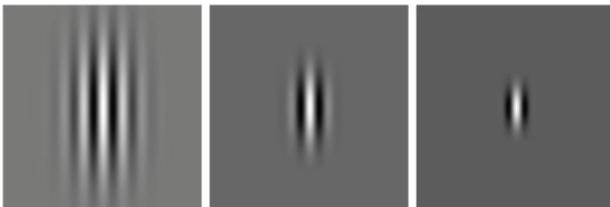


Рис. 1. Примеры фильтров Габора

Фильтр Габора является эффективным средством формирования локальных признаков цифрового изображения. Таким образом, проводя двумерную свертку изображения с Габоровским ядром сверки в точке x, y , получаем локальную характеристику этой точки. Так как фильтр Габора устойчив к операциям масштабирования и поворота, его можно использовать для задач идентификации пользователя. Для достижения этой цели необходимо выполнить несколько шагов.

1-й шаг: подготовка фильтра. Перед применением необходимо получить его ядро. В данной работе используется ядро фильтра размером 8×8 пикселей:

$$Filter[i + 4, j + 4] = g(x, y; \lambda, \theta, \psi, \sigma, \gamma), i \in [-4; 4], j \in [-4; 4].$$

2-й шаг: поиск опорных областей на изображении производится с помощью алгоритма Виолы — Джонса [1]. В качестве опорных областей были выбраны левый глаз, правый глаз, нос, рот.

```
faces = img.HaarDetectObjects(_cascadeMouth, _storage);
fac-
es = img.HaarDetectObjects(_cascadeRightEye, _storage);
fac-
es = img.HaarDetectObjects(_cascadeLeftEye, _storage);
faces = img.HaarDetectObjects(_cascadeNose, _storage);
```

3-й шаг: применение фильтра в опорных точках на изображении и получение вектора признаков, который формируется из значений фильтров Габора в опорных точках [2]. Значение фильтра Габора вычисляется с помощью операции свертки в точке $[x, y]$:

$$Img[x, y] = \sum_{i=x-n/2}^{x+n/2} \sum_{j=y-n/2}^{y+n/2} Img[i, j] * Filter[i - (x - n/2), j - (y - n/2)],$$

где Img — исходное изображение, n — ширина ядра фильтра.

В качестве опорных точек были выбраны углы и центры опорных областей, полученных на 2-м шаге. Вектор признаков вычисляется для каждого изображения в базе данных и хранится вместе с ним. Затем на основе этих векторов проводится процесс распознавания.

Для входного изображения также вычисляется вектор признаков, для которого поочередно находится евклидово расстояние до каждого вектора в базе данных. Подходящим изображением является то, которое обладает вектором признаков с наименьшим расстоянием до входного.

Блок-схема алгоритма распознавания с использованием фильтра Габора представлена на рис. 2.

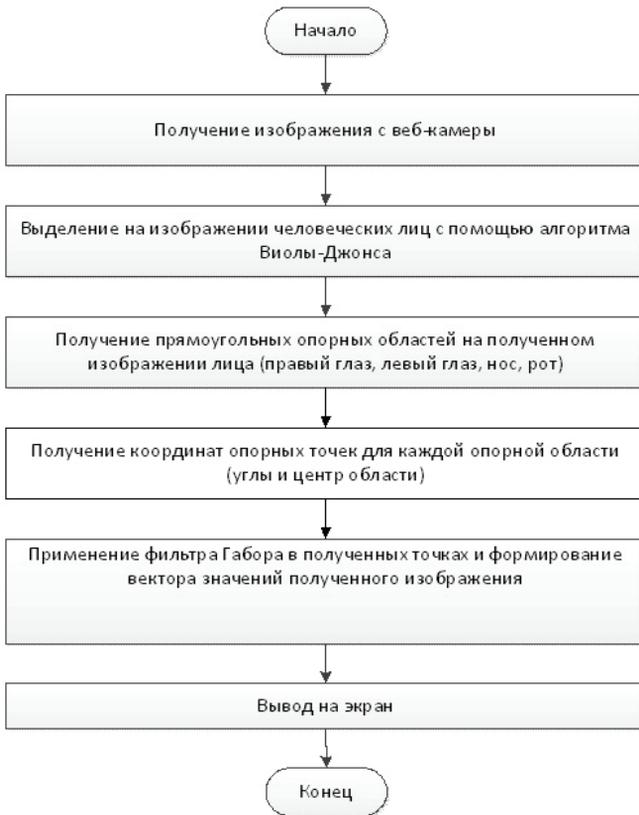


Рис. 2. Блок-схема алгоритма на основе фильтров Габора

Сравнение характеристик алгоритмов. Применимость алгоритмов распознавания лиц в различных ситуациях может быть оценена исходя из результатов проведенных исследований.

На рис. 3 и 4 представлены результаты исследования точности алгоритмов. Здесь точность понимается как отношение количества корректных результатов распознавания лица к общему количеству попыток. Алгоритмы тестировались для различных расстояний между камерой и пользователем. Алгоритм EigenFace более точен, причем с уменьшением расстояния до камеры его точность возрастает.

Скорость распознавания одного кадра зависит от количества фотографий в базе данных (рис. 5).

Полученные результаты показывают, что алгоритм с использованием фильтров Габора работает быстрее. Это связано с тем, что размерность анализируемого пространства в нем существенно меньше ($\dim 20$), чем в EigenFace ($\dim 10000$).

Важно отметить, что по причине большего количества вычислений для каждого сравнения алгоритм EigenFace очень чувствителен к объему базы данных, чего нельзя сказать об алгоритме с использованием фильтров Габора.

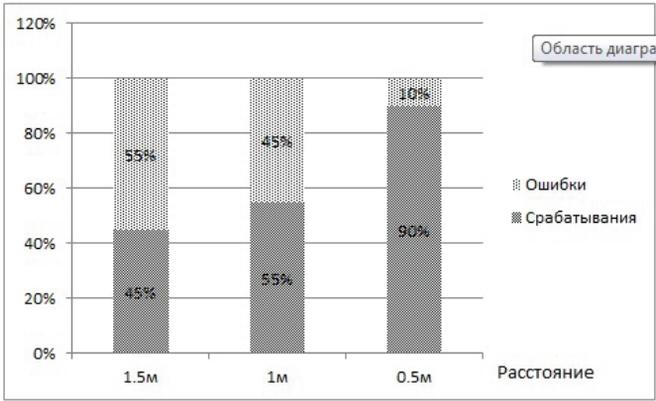


Рис. 3. Точность алгоритма EigenFace

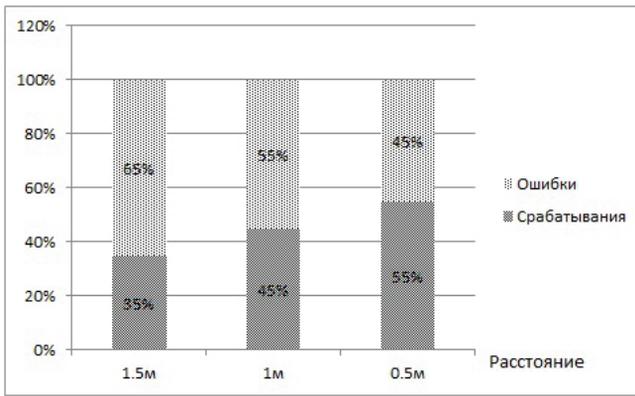


Рис. 4. Точность алгоритма с применением фильтров Габора

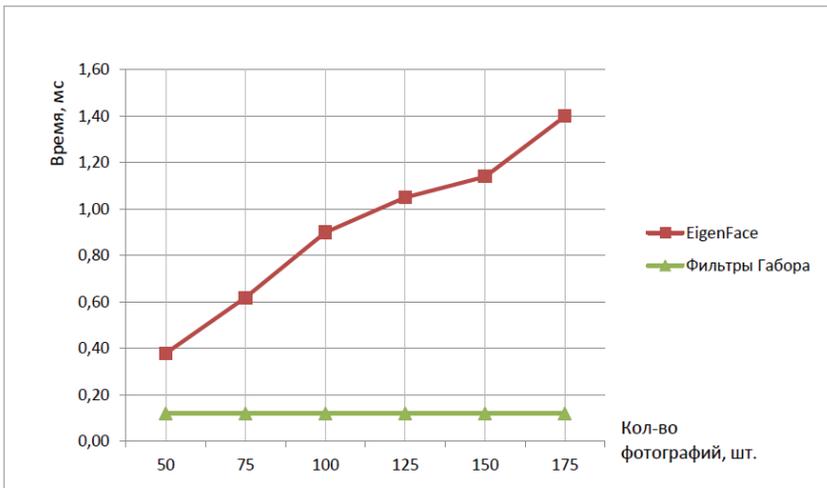


Рис. 5. Зависимость времени работы алгоритмов от количества изображений в базе данных

Значения показателей FAR (доля ложных срабатываний) и FRR (доля ложных отказов) для разработанной системы при использовании разных алгоритмов распознавания пользователя по изображению приведены в табл. 1.

Таблица 1

Сравнение значений показателей FAR и FRR

	FAR	FRR	Срабатывания
EigenFace	13%	6%	81%
Фильтры Габора	26,5%	0%	73,5%

Заключение. Для эксперимента использовалась система, в которой подходящими считались все результаты, не превышающие некоторого порогового значения. Такой подход позволяет получить выборку результатов вместо одного, что увеличивает вероятность совпадения. К тому же варьирование этого значения предоставляет возможность ужесточения либо ослабления критериев распознавания. Следовательно, скорость работы алгоритма с применением фильтров Габора позволяет рекомендовать его для распознавания лиц в больших потоках для первичного отбора. Однако недостаточная точность требует вторичной обработки отобранных изображений более точным алгоритмом, например EigenFace. В свою очередь, алгоритм EigenFace может успешно функционировать в системах особого контроля с малым потоком пользователей, так как требует значительных ресурсов.

ЛИТЕРАТУРА

- [1] Viola P. and Jones M.J. Robust real-time face detection. *International Journal of Computer Vision*, vol. 57, no. 2, 2004, pp. 137–154.
- [2] Гонсалес Р., Вудс Р. *Цифровая обработка изображений*. Москва, Техносфера, 2005, 1072 с.

Статья поступила в редакцию 10.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Рогозин О.В., Кладов С.А. Сравнительный анализ алгоритмов распознавания лиц в задаче визуальной идентификации. *Инженерный журнал: наука и инновации*, 2013, вып. 6 URL: <http://engjournal.ru/catalog/it/hidden/818.html>

Рогозин Олег Викторович — канд. техн. наук, доцент кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана. e-mail: logic00@mail.ru

Кладов Станислав Александрович — студент кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана. e-mail: stas@kladov.ru