

О ПРОСТРАНСТВЕННЫХ ХАРАКТЕРИСТИКАХ АЛГОРИТМА FINGERCODE

Аннотация

В работе рассматривается алгоритм дактилоскопической идентификации FingerCode и выделяются его пространственные характеристики. Производится постановка вычислительного эксперимента по установлению связи между указанными характеристиками и качеством распознавания алгоритма. Делается вывод о применении методов, основанных на FingerCode, для решения задачи биометрической идентификации.

Ключевые слова: отпечаток пальца, дактилоскопия, FingerCode, распознавание образов, биометрия, фильтры Габора.

ВВЕДЕНИЕ

Дактилоскопическая идентификация остаётся самым распространённым способом установления личности, применяемым в биометрии. Начиная с 1974 года, когда был впервые создан автоматизированный распознаватель (далее мэтчер), в ней существует ряд устоявшихся методов для распознавания изображений отпечатков пальцев. В частности, большинство современных систем основано на распознавании отпечатков путём сравнения множеств выделенных особых точек или минуций. Этот метод был исторически первым и за 35 лет получил развитие как в России, так и за рубежом [4, 9].

В то же время предлагались и предлагаются новые методы распознавания. Одним из наиболее перспективных методов является метод FingerCode, предложенный Прабакаром (под руководством проф. Джейна) [5, 7]. Этот метод основан на выделении опорной точки в отпечатке и анализе и срав-

нении статистических характеристик гребневого узора в её окрестности. Указанные публикации дали толчок к более детальному изучению этого метода.

Последующие работы установили, что качество распознавания с использованием FingerCode существенно зависит от точности нахождения опорной точки (например, [10]), в связи с чем были предложены новые методы её определения [3]. Сами авторы метода указывали, что скорость работы метода оставляет желать лучшего, и связывали это с имевшейся на тот момент аппаратной базой.

Однако ни в оригинальных, ни в последующих работах не проводилось обоснования пространственных характеристик окрестности опорной точки и их связи с качеством распознавания. Ответ на этот вопрос представляется нам важным, поскольку изменение пространственных констант может оказаться достаточным для серьёзного улучшения идентифицирующей способности алгоритма. В данной работе сделана попытка внести ясность в этот вопрос.

Работа структурирована следующим образом: сначала даются общие сведения о построении биометрических систем и описание алгоритма FingerCode, затем даются определения и исследуемые интервалы изучаемых пространственных констант, после чего формулируется вычислительный эксперимент и анализируются его результаты.

ОПИСАНИЕ АЛГОРИТМА FINGERCODE

Структурная схема биометрической системы, решающей задачу идентификации, представлена у Болла и др. в [1] (рис. 1). С точки зрения программной инженерии, это классический случай архитектуры «Фильтры и трубы» («Pipeline»). Следует также отметить, что биометрическая система всегда является частью подсистемы безопасности другой системы, которая выполняет бизнес-требования заказчика и не является предметом нашего рассмотрения.

Применительно к дактилоскопии, система работает следующим образом. Сканер считывает исходное изображение отпечатка пальца и передаёт его на экстрактор, который, возможно, улучшает характеристики исходного изображения (в частности, контрастность в случае недостаточно контрастного или некорректно настроенного FTIR-сканера) и строит по нему шаблон отпечатка. В случае с распознаванием по минуциям, например, этот шаблон представляет собой одномерный массив, в котором хранится информация о типе минуции (конец линии, ветвление), её направлении и координатах на изображении. Биометрическая БД хранит

шаблоны отпечатков зарегистрированных пользователей, которые вместе с шаблоном полученного отпечатка являются входной информацией для биометрического мэтчера. Мэтчер определяет наиболее похожие на предоставленный шаблоны в рамках данной стратегии распознавания и передаёт эти результаты в протокол принятия решения, который предоставляет доступ или отказывает в нём.

Алгоритм FingerCode [2] определяет реализацию экстрактора и мэтчера, а также структуру биометрической базы данных.

Экстракция свойств происходит в два этапа:

- 1) поиск опорной точки;
- 2) построение шаблона.

Первый этап начинается с построения векторного поля направлений папиллярных линий отпечатка пальца. Предлагается использовать метод, описанный в [8]. Использование матрицы Собеля и сглаживания направлений усредняющим фильтром даёт качественный результат, обладая при этом высокой скоростью работы. Хотя авторы предлагают в качестве более требовательного к вычислительным мощностям, но при этом более эффективного метода выделения папиллярных линий, метод Марра-Хилдретта, мы считаем, что целесообразнее использовать алгоритм Канни как более эффективный.

После построения поля направлений по нему осуществляется поиск опорной точки. Для этого для каждого пиксела (x_0, y_0) производится вычисление следующей метрики:

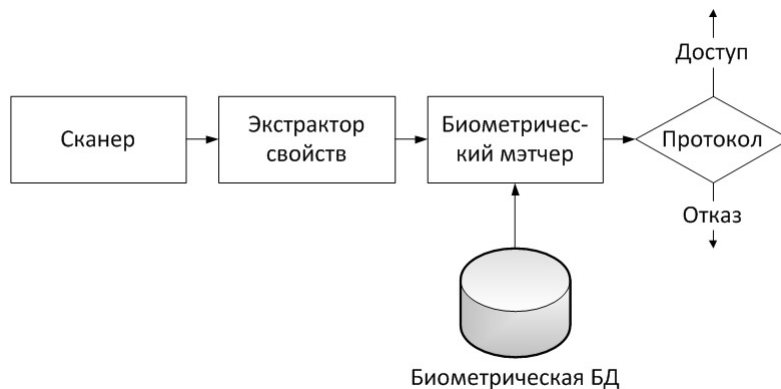


Рис. 1. Общая схема биометрической системы

$$K = \sum_{II} \sin(O(x, y)) - \sum_I \sin(O(x, y)).$$

Здесь I и II – области вокруг пиксела (x_0, y_0) , вхождение в которые соседних с данным пикселем определяется следующими соотношениями:

$$\forall (x, y) \in I: (x - x_0)^2 + (y - y_0)^2 \leq r^2,$$

$$y \geq y_0, \left| \arctg \left(\frac{y - y_0}{x - x_0} \right) \right| < 1,$$

$$\forall (x, y) \in II: (x - x_0)^2 + (y - y_0)^2 \leq r^2,$$

$$y \geq y_0, \left| \arctg \left(\frac{y - y_0}{x - x_0} \right) \right| \geq 1.$$

Здесь r – некоторый радиус, $O(x, y)$ – угол наклона папиллярных линий в данной точке, находящийся в пределах от 0 до π . Авторы предлагают итерационный процесс нахождения опорной точки. Сначала метрика вычисляется по всему изображению, и в качестве первого приближения выбирается пиксел с максимальным значением. Последующие шаги вычисляют метрику для меньшего радиуса в окрестности текущего приближения. Наши эксперименты показали, что такой способ нахождения опорной точки не эффективен, если отпечаток пальца оказывается повернут уже на 10–20 градусов. В качестве альтернативы предлагается использовать индекс Пуанкаре (например, в [6]), однако мы считаем, что описываемый в [4] алгоритм нахождения более прост и так же надёжен при сопоставимых вычислительных затратах.

После нахождения опорной точки строится шаблон сравнения. Область вокруг опорной точки разделяется на M концентрических колец, разделённых на N секторов. При этом ширина колец одинакова и составляет K пикселей, а радиус окружности внутри наименьшего из них составляет L пикселей. Авторы алгоритма предлагают $M = 5, N = 16, K = 20, L = 20$.

Сначала проводится нормализация секторов колец к одним и тем же значениям математического ожидания и дисперсии цвета пикселей. Цвет каждого пиксела в каждом секторе определяется как

$$C_i(x, y) = \begin{cases} M_0 + \sqrt{\frac{D_i(C(x, y) - M_i)^2}{D_i}}, & C(x, y) > M_i, \\ M_0 - \sqrt{\frac{D_i(C(x, y) - M_i)^2}{D_i}}, & C(x, y) \leq M_i, \end{cases}$$

где M_0 и D_0 – желаемые значения математического ожидания и дисперсии, которые у авторов метода установлены в 100. После нормализации вектора над каждым пикселем производится свёртка посредством P фильтров Габора. Симметричный фильтр Габора в двумерном пространстве описывается формулой

$$G(x, y, f, \theta) = e^{-0.5 \left(\frac{x'^2}{\delta_{x'}^2} + \frac{y'^2}{\delta_{y'}^2} \right)} \cos(2\pi f x'),$$

$$x' = x \sin \theta + y \cos \theta,$$

$$y' = x \cos \theta - y \sin \theta,$$

где f – частота синусоидальной волны, направленной по углу θ и, считая от оси x , $\delta_{x'}$, $\delta_{y'}$ – пространственные константы гауссовой свёртки по осям x' и y' . Размер маски фильтра определяется авторами метода как 33×33 , частота f как $1/D$, где D – среднее расстояние между папиллярными линиями в пикселах, что для изображения 500 dpi составляет 10 пикселей, $\delta_{x'} = \delta_{y'} = 4.0$ – эмпирически подобранная авторами константа, равная примерно половине расстояния между папиллярными линиями. Углы фильтров определяются как

$$\theta = \frac{i\pi}{P}, i = \overline{0..P-1}.$$

Авторы метода устанавливают $P = 8$.

По пикселям, попадающим в кольца, формируется шаблон. Для каждого сектора вычисляется среднее отклонение

$$D_{i\theta} = \frac{\sum_{n_i} |F_{i\theta}(x, y) - M_{i\theta}|}{n_i},$$

где n_i – число пикселей в секторе, $F_{i\theta}(x, y)$ – значение цвета данного пиксела с координатами (x, y) , $M_{i\theta}$ – математическое ожидание цвета пиксела в данном секторе. Мы обратили внимание, что количество пиксе-

лов в кольцах, которые необходимо подвергнуть обработке, определяется как

$$N_{\text{пикс}} = N(L + MK) - N(L).$$

$N(r) = 1 + 4 \sum_{i=0}^{\infty} \left(\left\lfloor \frac{r^2}{4i+1} \right\rfloor - \left\lfloor \frac{r^2}{4i+3} \right\rfloor \right)$ – количество целых точек внутри окружности данного радиуса. В случае авторских параметров $N(120) - N(20) = 45225 - 1257 = 43998$ пикселей для каждого фильтра. Необходимо отметить, что с ростом $MK + L$ число пикселей, подлежащих обработке, нелинейно увеличивается, что приводит к замедлению работы алгоритма. На наш взгляд, изучение влияния M , K и L на качество распознавания может также привести и к ускорению работы. Пример сформированного FingerCode показан на рис. 2.

Сопоставление отпечатков пальцев осуществляется путём вычисления евклидовой меры между двумя сравниваемыми шаблонами, каждый из которых представляется в

виде многомерного вектора. Полученная мера является мерой идентичности отпечатков. Они тем более похожи, чем меньше её значение.

Каждый отпечаток, таким образом, представляется вектором из $M \cdot N \cdot P$ целых чисел. Для стандартного 4-байтового целого, оригинальный FingerCode занимает $5 \cdot 16 \cdot 8 \cdot 4 = 2560$ байт. Не сложно посчитать, что база из 10000 отпечатков будет занимать ровно 25 Мб, однако в этом расчёте не учитывается дополнительная информация, связанная с отпечатком пальца, в частности, идентификатор пользователя.

К явным достоинствам этого алгоритма можно отнести очень быструю процедуру сравнения, относительно небольшой размер шаблона. К недостаткам алгоритма следует отнести длительное время построения шаблона и низкую точность. Следует отметить, что первый недостаток не является принципиальным. Как на этапе поиска опорной точки, так и на этапе построения шабло-

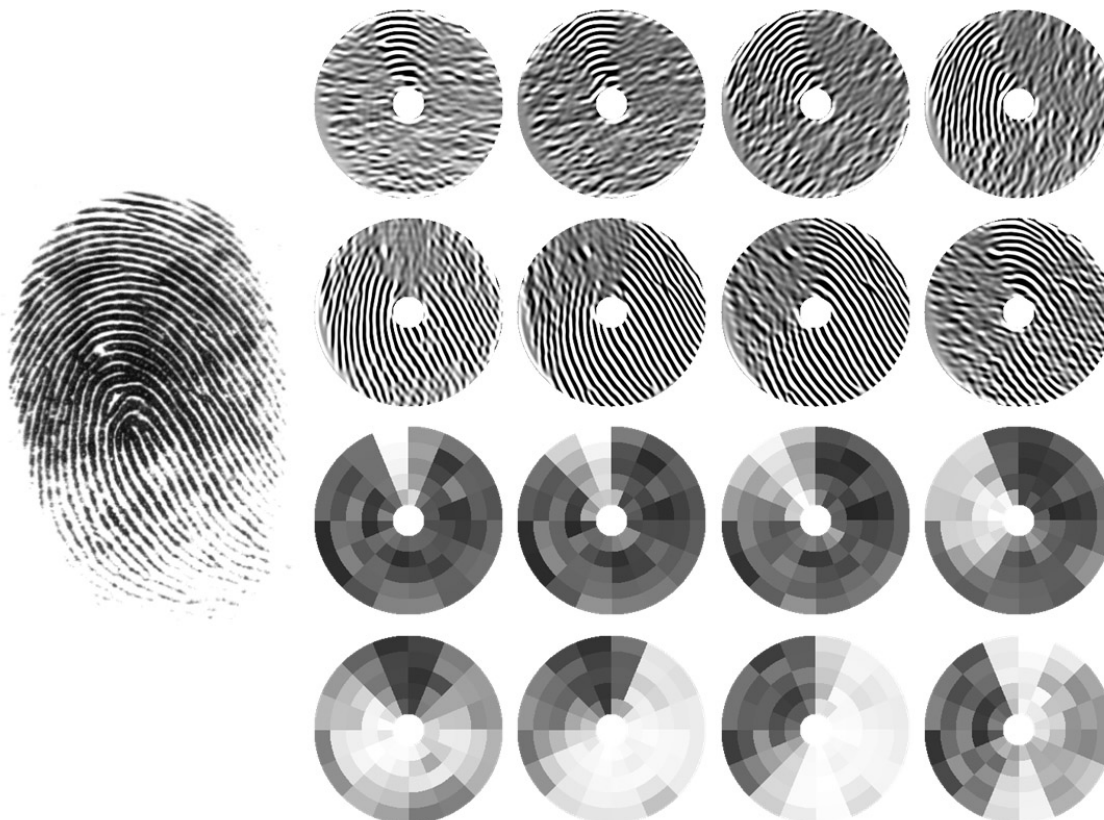


Рис. 2. Исходный отпечаток пальца, 8 отфильтрованных изображений и соответствующие им компоненты вектора FingerCode (дисперсия тем больше, чем больше сектор)

на итерации во всех итеративных процессах могут выполняться независимо друг от друга, что открывает большие возможности по распараллеливанию алгоритма. Нам представляется наиболее перспективным применение технологий семейства GPGPU. С другой стороны, сами авторы метода предлагают эффективную эвристику, согласно которой из фильтра Габора удаляются все элементы, не превышающие по модулю 0.05. Наши эксперименты показали, что, благодаря этому, время фильтрации изображения сокращается в 4 раза.

ПОСТАНОВКА ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

Как мы отмечали ранее, в предыдущих работах, связанных с алгоритмом FingerCode, не было дано обоснования для выбора числа колец, числа фильтров, числа секторов в кольце, толщины колец и внутреннего радиуса наименьшего из них. Таким образом, не установлена связь между качеством распознавания и пространственными константами.

Качество алгоритма распознавания принято оценивать в терминах FAR (False Accept Rate, отношение некорректных предоставлений доступа к общему числу распознаваний) и FRR (False Reject Rate, отношение некорректных отказов доступа к общему числу распознаваний). В работах авторов метода получены рабочие точки в 1.07 % и 7.87 %, а также 4.59 % и 2.83 %. Эти показатели являются низкими (в частности, для современных биометрических систем типовыми являются значения FAR = 10^{-6} , FRR = 10^{-5}), что может быть связано как с неточным определением опорной точки, так и с выбором пространственных констант M, N, K, L, P .

Наш вычислительный эксперимент заключается в оценке FAR и FRR для одной и той же базы отпечатков для разных M, N, K, L, P . Чтобы исследовать связь пространственных констант алгоритма с качеством распознавания, мы будем осуществлять сравнение шаблонов для изображений отпечатков с предварительно определёнными вручную опорными точками. Это позволит определить качественные характеристики

FingerCode без искажений со стороны алгоритма определения опорной точки. В нашем эксперименте мы используем множество В базы данных 1 FVC2000, состоящее из 80 отпечатков (по 8 отпечатков каждого из 10 пальцев) в tiff-файлах размером 480×640 и разрешением 500dpi. Базы данных FVC используются для объективного сравнения различных алгоритмов распознавания и составляются специально таким образом, чтобы включать сложные для различения наборы отпечатков. Для каждого набора параметров выбирается то множество изображений, на которых область колец не выходит за пределы изображения. Каждый отпечаток из этого множества сравнивается со всеми остальными, и определяется тот, для которого евклидова мера минимальна. Нужно отметить, что порог в FingerCode определяется отлично от классического определения в [1]. Поскольку чем меньше значение меры, тем более похожи отпечатки, процедура сравнения с порогом инвертирована. Если мера больше пороговой, ситуация рассматривается как ложный отказ. Если мера меньше пороговой, но при этом отпечатки принадлежат разным пальцам, это случай ложного доступа. Отношение случаев ложного отказа и ложного доступа к общему числу сравнений есть FRR и FAR соответственно. При этом с ростом порога нелинейно растёт число ложных доступов и FAR, а FRR уменьшается. С уменьшением порога ситуация обратная: FAR снижается, а FRR растёт. Значение порога выбирается таким образом, чтобы обеспечить максимально низкий FAR при достаточно удобном для пользователей FRR.

Выбор порога является существенным для нашего эксперимента. Максимальное расстояние для вектора из R компонент определяется как $(V_{max} - V_{min}) \sqrt{R}$. Для значений цвета, лежащих в диапазоне $[0; 255]$, $V_{max} = 128$, $V_{min} = 0$. Минимальное расстояние равно 0. Для нашей тестовой базы отпечатков мы провели эксперимент с пространственными константами, установленными в значениях авторов метода, для диапазона пороговых значений $[0.05; 0.5]$ с шагом 0.05. Часть данных приведены в табл. 1, полная

версия данных доступна по адресу <https://docs.google.com/open?id=0B5b8aIRQusHrMDVINmNIODUtYmI3MC00NGE2LWlZ N2QtY2JkZDFjMzViODdh>. Оптимум находится в районе точки 0.2 с FAR = 10.39 % и FRR = 5.19 %. Значение 0.2 мы используем далее как константу порога.

Определим границы интервалов для каждой константы.

Число фильтров P в нашем эксперименте лежит в диапазоне от 4 до 8. В некоторых работах ([6]) предлагается наращивание банка фильтров до 16, однако нам это представляется нецелесообразным. Дело в том, что фильтрами Габора выделяются не только участки линий, угол наклона которых к горизонтальной прямой равен и фильтра, но и участки с наклоном в диапазоне порядка $[\theta - K, \theta + K]$, $K \in \left[\frac{\pi}{8}, \frac{\pi}{6} \right]$. Поэтому увеличение числа фильтров приводит к перекрыванию областей с папиллярными линиями на преобразованных изображениях и увеличению времени фильтрации. С другой стороны, 3 и менее фильтров недостаточно для выделения всего множества папиллярных линий на рисунке.

Число колец M выбирается в пределах от 3 до 5, так как меньше 3 колец недостаточно для описания статистических характеристик папиллярных линий отпечатков, а больше 5 – избыточно и будет приводить к зашумленности колец, начиная с шестого, вследствие попадания в исследуемую область участков без папиллярных линий. Толщина кольца K и внутренний радиус наименьшего из них L выбираются в пределах от 10 до 30 пикселей, что соответствует размещению в одном кольце от 1 до 3 папиллярных линий для изображений в тестовой базе. Необходимо отметить, что расстояние между папиллярными линиями, измеренное в пикселях, является основополагающим параметром для выбора диапазона. Меньший размер кольца приведёт к тому, что статистические характеристики секторов не будут коррелировать со статистическими характеристиками изображения, а больший – к попаданию в исследуемую область за преде-

Табл. 1. Экспериментальные результаты FAR и FRR для различных значений порога при оригинальных структурных характеристиках FingerCode

Порог	FRR	FAR
0,18	0,207792	0,051948
0,185	0,181818	0,051948
0,19	0,155844	0,064935
0,195	0,077922	0,090909
0,2	0,051948	0,103896
0,205	0,051948	0,103896
0,21	0,051948	0,103896
0,215	0,025974	0,12987

лом отпечатка, которая, как и в случае с M , бесполезна для распознавания. Число секторов N выбирается в пределах от 8 до 16, то есть равно удвоенному диапазону P . Увеличение количества секторов линейно увеличивает время сравнения, однако более важным фактором является тот факт, что шаблон, полученный из разбиения на большое число секторов, больше подвержен шумам. С другой стороны, при малом N возникает ситуация, когда области с выделенными линиями и без них объединяются воедино, вследствие чего значение дисперсии оказывается искажено большой долей областей без линий относительно областей с линиями.

Ещё одним важным моментом является быстродействие. Пусть $I(x, y)$ – исходное изображение, тогда нормализованное изображение $IN(x, y) = f(I(x, y), M, K, N, L)$, а изображение, преобразованное i -м фильтром Габора $FB(i)$, тогда можно представить как

$$IG(x, y) = g(IN(x, y), FB(i), M, K, L) = g'(I(x, y), FB(i), M, K, N, L).$$

Поскольку $N \in [8, 16]$, процедура фильтрации P фильтрами осуществляется N раз для каждого набора (M, K, L) .

Исходный код программы для нашего эксперимента написан на языке программирования C# с целевым рантаймом .NET 4.0 с использованием IDE MS Visual Studio 2010. Этот и другие эксперименты в нашей работе проводились на следующей конфигурации: Intel Core i5-2400 с частотой 3.3 ГГц,

8 Гб ОЗУ, материнская плата Gygabyte GA-Z68MA-D2H-B3 (частота работы с ОЗУ 2133 МГц), операционная система Windows 7 Ultimate x64. Программа запускалась в режиме debug, и, по нашим оценкам, её время работы составляло 29 дней. Поэтому мы рассмотрели вариант, при котором нормализация подвергается сразу вся площадь колец без деления на секторы. Такой подход способствует повышению быстродействия, так как тогда нормализация перестаёт зависеть от N , а значит, время работы уменьшается приблизительно в 12 раз.

Пусть $IN(x, y)$ – цвет пиксела изображения, которое нормализовали отдельно для каждого сектора, $IN'(x, y)$ – цвет пиксела изображения, кольца в котором были нормализованы как единый сектор. Мы провели эксперимент, где для тестовой базы и наборов (M, K, L) вычислялось среднее значение модуля $|IN(x, y) - IN'(x, y)|$. Этот показатель позволяет оценить, как сильно изменяется изображение при применении нормализации по всем кольцам. Также мы оценили, сколько пикселей в кольцах изменило свой цвет более чем на 12,75 (5 % длины диапазона $[0; 255]$) и сколько пикселей изменило свой цвет более, чем на 5 % от первоначального. Эти значения показывают, насколько велика часть изображения, которая сильнее всего затрагивается нормализацией.

Результаты этих вычислений находятся по адресу [https://docs.google.com/open?id=](https://docs.google.com/open?id=0B5b8aIRQusHrNTQxNzRkOGYtZDFmYy00ZDMyLWFkNWEtNjgzODMwOTU1NGJh)

[0B5b8aIRQusHrNTQxNzRkOGYtZDFmYy00ZDMyLWFkNWEtNjgzODMwOTU1NGJh](https://docs.google.com/open?id=0B5b8aIRQusHrNTQxNzRkOGYtZDFmYy00ZDMyLWFkNWEtNjgzODMwOTU1NGJh).

Данные приводятся для всех подходящих для распознавания отпечатков в порядке уменьшения среднего изменения цвета.

Средние значения метрик от $MK + L$ приведены на рис. 3–5. Анализ результатов показал, что увеличение всех трёх метрик коррелирует с ростом $MK + L$.

В частности, максимальное среднее изменение цвета пиксела составило 19.4 при изменении более чем на 5% цвета порядка 30% пикселей. Эти значения зафиксированы для случая 5 колец толщиной в 30 пикселей и радиусе внутренней окружности в 29 пикселей. Средние значения оказываются существенно меньше, что говорит о существенном разбросе значений. Мы объясняем это тем, что при малых M, K, L окружность такого радиуса с центром в опорной точке содержит только области с папиллярными линиями, статистические характеристики пикселей которой примерно одинаковы на всей площади. Однако с ростом радиуса захватываются и области без папиллярных линий – в границы окружности попадает область за пределами отпечатка пальца. Эта область обладает иными статистическими характеристиками, поэтому общий результат искажается. Следовательно, верхняя граница диапазона интервала $MK + L$ не может быть использована на практике для надёжного распознавания, так как границы отпечатка одного и того же пальца меняются при

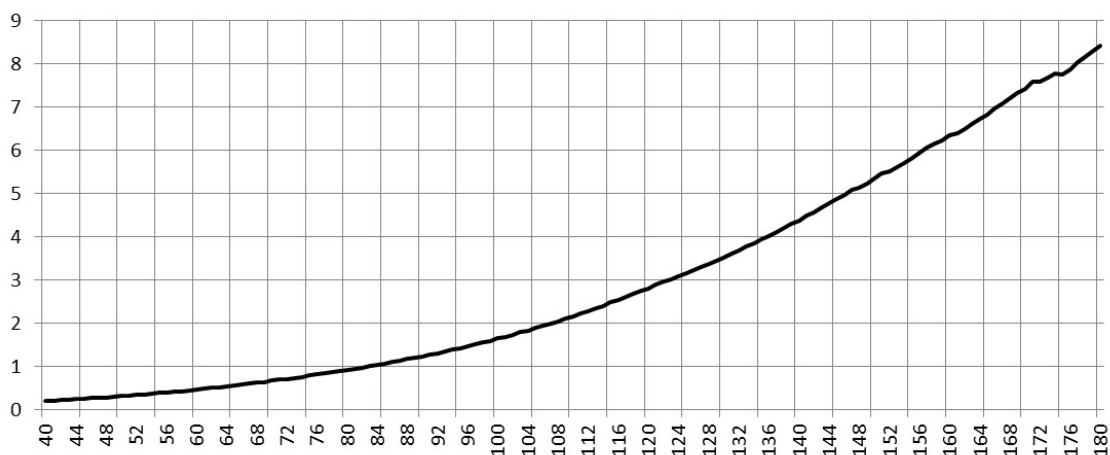


Рис. 3. Зависимость среднего изменения цвета от $MK + L$

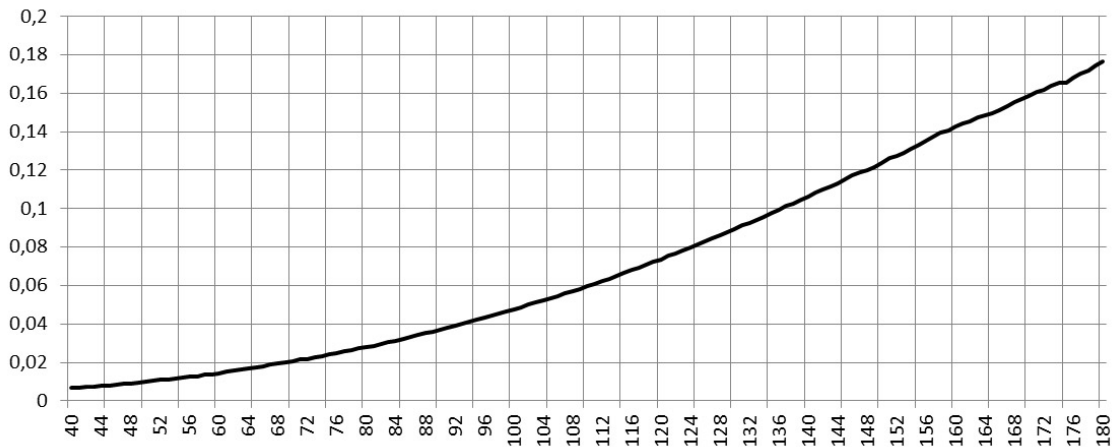


Рис. 4. Зависимость усреднённого процента пикселей, чьё изменение цвета превысило 12.75, от $MK + L$

каждом сканировании, а значит, меняются и статистические характеристики нормализуемой области. Мы также считаем, что использование посекторной нормализации в данном случае приведёт к тому, что секторы отпечатков одного и того же пальца, находящиеся ближе к центру колец, будут обладать в среднем меньшей разницей дисперсий, а более удалённые – большей. Надёжное распознавание не будет обеспечено и в этом случае.

При этом для малых (M , K , L) изменение цвета составляет десятые доли пиксела, а количество пикселей, существенно изменивших свой цвет, не превышает 1–2 %. Средние значения (M , K , L) характеризуют-

ся изменением цвета в среднем на 2–3, а число существенно изменивших свой цвет пикселей составляет около 10%, что тоже является приемлемым результатом, так как изменения такого уровня могут произойти и естественным путём при сканировании отпечатка, особенно на FTIR-сканерах. Поэтому мы считаем, что для вычислительного эксперимента было допустимо использовать нормализацию по всей площади колец. Её применение позволило снизить время расчёта до 2.5 дней.

Итоговый текст программы доступен по адресу <https://docs.google.com/open?id=0B5b8aIRQusHrZGFIMTQ3ZmMtZGNmNS00YTl3LWFiZTAzMzI4M2Y0ZmUyY2Q2>.

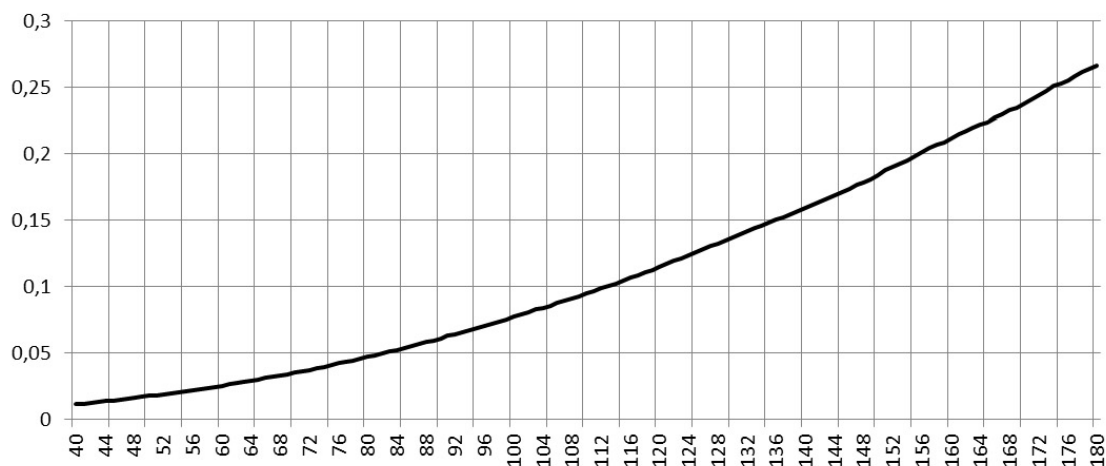


Рис. 5. Зависимость усреднённого процента пикселей, чьё изменение цвета превысило 5% от его первоначального значения, от $MK + L$

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА И ВЫВОДЫ

Таблица с результатами вычислительно-го эксперимента доступна по адресу <https://docs.google.com/open?id=0B5b8aIRQusHrMTQwMGM4ZmItNWQ4Yy00MmUxLTlmNDMtODc3NjQ0OTg1MDcz>.

С точки зрения безопасности показатель FAR считается важнее FRR, так как ситуация пропуска неавторизованного лица в охраняемую область намного хуже, чем требование повторной идентификации в случае ложного отказа [1]. Для применимости в идентификационных системах мы будем оценивать результаты по метрике $FRR + 10FAR$. В силу достаточно малого размера тестовой базы, мы будем оценивать лучшие 100 результатов и округлим их до ближайшего целого. Мы обнаружили, что для выбранной метрики $M = 4$ (3.57), $N = 15$ (14.74), $K = 12$ (11.95), $L = 10$ (10.25), $P = 6$ (5.71). При этом $FAR = 5.06\%$, $FRR = 8.16\%$. На основании этих данных мы делаем вывод, что даже при точном позиционировании опорной точки точности метода недостаточно для использования в целях идентификации.

Сдругой стороны, метрика $10FRR + FAR$ позволяет оценить применимость метода для фильтрации базы от отпечатков, не похожих на предоставленный, что можно использовать для создания многоуровневого или комбинированного мэтчера. В среднем, $M = 3$ (3.1), $N = 9$ (8.6), $K = 24$ (23.57), $L = 14$ (14.47), $P = 7$ (6.72). При этом $FAR = 12.26\%$, $FRR = 1.3\%$. Мы утверждаем, что метод применим для фильтрации базы, хотя для окончательного подтверждения этого утверждения нужна проверка на базе большего объёма.

Нужно отметить, что максимум в обоих случаях достигнут для числа колец, меньшего 5, что, на наш взгляд, указывает на избыточность и зашумленность пятого кольца. Радиус внутренней окружности также оказался меньше, чем в оригинальном алгоритме. Мы объясняем это тем, что папиллярные линии на расстоянии от 10 пикселей (одного расстояния между папиллярными линиями) образуют достаточно длинные отрезки на отфильтрованных изображениях,

чтобы использовать их для распознавания отпечатков. Число фильтров в первом случае оказалось относительно равномерно распределено в результатах с преобладанием чисел в верхней части диапазона. Во втором случае эта тенденция проявилась более явно. Количество секторов также оказывается чаще в верхней части диапазона. На основании этого мы полагаем, что большее число секторов и фильтров точнее описывают отпечаток.

Разница в ширине кольца в первом и втором случае мы объясняем разницей критериев оптимизации. В первом случае более тонкие кольца содержат информацию только об области вокруг опорной точки, и её достаточно для идентификации. Во втором случае кольца большей ширины не позволяют точно отличить один отпечаток от другого, однако в шаблоне содержится больше информации об общем узоре гребневых линий, что позволяет отсеять непохожие отпечатки.

ЗАКЛЮЧЕНИЕ

Мы провели изучение пространственных характеристик алгоритма FingerCode и получили данные, которые свидетельствуют об их существенном влиянии на качество распознавания: их корректный выбор позволяет значительно улучшить показатели FAR и FRR. К сожалению, как изначальная, так и модифицированные версии алгоритма неприменимы для дактилоскопической идентификации в промышленных системах из-за недостаточно точного распознавания. Однако достигнутые на текущий момент временные характеристики, а также адаптированные под текущую задачу константы позволили нам достичь значений FRR и FAR, которые говорят о приемлемости основанных на FingerCode методов в биометрических системах, использующих слияние алгоритмов на различных уровнях [5]. В частности, нам представляется перспективной двухступенчатая идентификация, где первоначальная фильтрация банка отпечатков осуществляется FingerCode-подобным алгоритмом, а точное распознавание производится методом, основанным на минуциях.

Наши дальнейшие исследования направлены на улучшение временных характеристик и построение комбинированного распознавателя.

Литература

1. *Bolle Ruud u др.* Guide to Biometrics. London : Springer Professional Computing, 2003.
2. *Jain Anil K. u др.* Filterbank-Based Fingerprint Matching // IEEE Transactions on Image Processing. Piscataway, N.J. : IEEE Signal Processing Society, May 2000 г. 5 : Т. 9.
3. *Jain Anil K., Prabhakar Salil u Hong Lin.* A Multichannel Approach to Fingerprint Classification // IEEE Transactions on Pattern Analysis and Machine Intelligence. Washington, DC : IEEE Computer Society, 1999 г. 4 : Т. 21.
4. *Lifeng Sha Feng Zhao, Xiaoou Tang.* Improved FingerCode for Filterbank-based Fingerprint Matching // Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on . Washington, DC : IEEE, 2003.
5. *Maltoni Davido u др.* Handbook of Fingerprint Recognition. London : Springer-Verlag London Limited, 2009. 2nd.
6. *Munir Muhammad Umer, Javed и Younas Muhammad.* Fingerprint Matching using Gabor Filters // National Conference on Emerging Technologies Proceedings. Karachi : IEEE Karachi, 2004.
7. *Prabhakar Salil.* Fingerprint Classification and Matching Using, Ph.D. thesis. East Lansing, MI : Michigan State University, 2001.
8. *Ratha N. K., Chen S. Y., Jain A. K.* Adaptive flow orientation-based feature extraction in fingerprint images // Pattern Recognition. Burlington, MA : Elsevier Academic Press, 1995 г. 11 : Т. 28.
9. *Гудков В.Ю.* Способы математического описания и идентификации отпечатков пальцев // Труды ИСА РАН. Москва : Учреждение Российской академии наук Институт системного анализа Российской академии наук, 2008 г. - Т. 38.
10. *Шафранская А.Я., Тузиков А.В.* Алгоритм сопоставления отпечатков пальцев // Сборник тезисов докладов X Республиканской научной конференции студентов и аспирантов высших учебных заведений Республики Беларусь «НИРС-2005». Минск : Отдел НИРС Управления подготовки кадров высшей квалификации Главного управления науки БГУ, 2005. Т. 2.

Abstract

The fingerprint recognition algorithm FingerCode is evaluated and its spatial characteristics are observed. The computational experiment on establishing the connection between those characteristics and algorithm recognition quality is proposed. The conclusion is made for FingerCode-based algorithms to be used in the biometric identification.

Keywords: fingerprint, fingerprinting, FingerCode, image recognition, biometrics, Gabor filters.

*Сартасов Станислав Юрьевич,
аспирант математико-
механического факультета СПбГУ,
разработчик «Ланит-Терком»,
Stanislav.Sartasov@gmail.com
Stanislav.Sartasov@lanit-tercom.com*



Наши авторы, 2011.
Our authors, 2011.