

Алгоритмы поиска оптимального положения образов при их суммировании

Аннотация. В статье рассматриваются вопросы совмещения нескольких изображений одного объекта и получения суммарного (усредненного) образа. На простых примерах показывается, что такое усредненное изображение само по себе не может служить при распознавании хорошим эталоном для сравнения. Предлагается другое использование суммарного образа для распознавания. Приводятся эксперименты с его построением различными методами и оценка их быстродействия.

Ключевые слова: изображение, функция близости, оптимальное положение фигур, расстояние Хэмминга, Манхэттенское расстояние, дискретное преобразование Фурье.

Введение

В задачах распознавания постоянно приходится иметь дело с объектами абсолютно одинаковой природы или разными изображениями одного объекта.

Имея несколько изображений одного объекта, можно попытаться совместить их и получить суммарный (или усредненный) образ объекта. В этом образе устранено влияние признаков, не являющимися общими для всех изображений. Суммарный образ может служить как источником эталонов для распознавания, так и для других целей, например, для более качественного воспроизведения изображения объекта.

Будем представлять изображенные объекты плоскими, так что их элементы всюду одни и те же на всех изображениях. Примерами таких объектов являются части бумажного документа: буквы, цифры, печати, баркоды, штампы и т.п. Также это могут быть элементы разметки дорожного полотна, на нескольких фотографиях, снятых с одного и того же места.

Со сканера или фотокамеры изображения могут получаться цветными, полутоновыми (серыми) или черно-белыми (бинарными), однако в данной работе не будут рассматриваться цветные изображения, поскольку в образах до-

кументов большинство объектов являются однотонными.

С другой стороны, даже если исходные объекты черно-белые, после совмещения нескольких экземпляров и, например, усреднения, получим полутоновое изображение. Оно ничем не отличается от того, как если бы исходные картинки были бы серыми. Поэтому там, где это не оговорено особо, будем предполагать изображения полутоновыми.

Говоря о нескольких изображениях одного объекта, будем рассматривать различные ситуации. В одной из них объекты исходно являются физически разными как буквы «а» на странице текста. В другой – это разные изображения одного объекта, например, печать на десяти фотографиях одного документа. Однако большой разницы в этих ситуациях нет. Заметим, что интерес к проблеме совмещения большого количества идентичных объектов обусловлен именно возможностью получения в видеопотоке многократного воспроизведения одного объекта.

Поскольку на фотоснимках практически всегда отсутствует абсолютная система координат, проблема отождествления объекта на разных снимках часто представляет большую сложность. Но даже если сами объекты выделены и представлены в виде матрицы пикселей,

«правильное» совмещение их является важной и не всегда тривиальной задачей.

Во многих алгоритмах распознавания производится сравнение двух образов с помощью некоторой функции близости, которая выбирается так, чтобы удовлетворять требованиям алгоритма распознавания, и учитывать особенности образов, получаемых из заданного источника.

Рассмотрим пространство образов U , представленных в виде матриц $R = \|r_{ij}\|$ одинакового размера $0 < i \leq m$, $0 < j \leq n$, где $r_{ij} \in \{0, 1\}$ (бинарный образ), либо $r_{ij} \in \{0, 1, \dots, 2^{p-1}\}$ (полутоновой образ). Размеры реального образа существенно меньше матрицы $m \times n$. Размещение образа с $m_1 \times n_1$ ($m_1 < m$, $n_1 < n$) назовем *центрированием*. Если реальный образ представлен матрицей $m_1 \times n_1$, то можно, например, центрировать этот образ так, чтобы центры матриц $m_1 \times n_1$ и $m \times n$ совпадали. В работе [1] предложен иной способ центрирования, состоящий в использовании центра масс. Можно придумать и применять другие способы центрирования, для нас важно, что способ центрирования один для всех образов.

При получении образов символов с помощью цифровых устройств (сканеры, фото- и видеокамеры) возникает ряд *эффектов оцифровки*, главный из которых состоит в том, что из одного прообраза могут получаться образы, различающиеся как между собой, так и прообразом.

Среди других эффектов оцифровки отметим искажения, связанные с наклоном, и искажения, происшедшие от случайного шума. При получении образа с камеры к этому добавляются эффекты изменения освещения.

К функции близости d , используемой в кластеризации и распознавании, предъявляются два требования:

- аксиома тождества - $d(A, B) = 0 \Leftrightarrow A = B$;
- аксиома симметрии - $d(A, B) = d(B, A) \forall A, B$.

Вклад эффектов оцифровки при сравнении образов минимизируется выбором функции близости и применением процедуры сдвига. В настоящей работе в качестве функции близости рассматривается «манхэттенское расстояние» (или метрика Хэмминга для бинарных образов):

$$\mu_0(A, B) = \sum_{i=1}^m \sum_{j=1}^n |a_{ij} - b_{ij}|.$$

При сравнении образов A и B из пространства U будем производить несколько *сдвигов* образа $A = \|a_{ij}\|$ в разных направлениях, выбирая в каче-

стве расстояния между образами A и B минимальную величину из полученных значений:

$$d_0(A, B) = \min(\mu_0(A^{(h,v)}, B)),$$

$$a_{ij}^{(h,v)} = a_{(i+v) \bmod m, (j+h) \bmod n}.$$

Рассмотрим другую функцию близости $d_1(A, B)$, основанную на псевдометрике $\mu_1(A, B)$. Для каждого образа $R \in U$ определим расширенный образ $O^{(1)}(R)$, включающий сам образ R и его единичную окрестность [2]. Заданную функцию близости между образами следующим образом:

$$\mu_1(A, B) = \sum_{i=1}^m \sum_{j=1}^n (a_{ij} \cdot \overline{O}_{b,ij}^{(1)} + b_{ij} \cdot \overline{O}_{a,ij}^{(1)}),$$

где $O^{(1)}(A) = \|O_{a,ij}^{(1)}\|$, $O^{(1)}(B) = \|O_{b,ij}^{(1)}\|$ – расширенные образы A и B .

Соответственно, расстояние в этой метрике

$$d_1(A, B) = \min_{(h,v)} (\mu_1(A^{(h,v)}, B)), \quad (1)$$

где $A^{(h,v)}$ – по-прежнему модифицированный образ A (параллельно сдвинутый на вектор (h, v)).

Если способ измерения расстояния задан, для суммирования образов надо рассмотреть несколько вопросов:

- как получить суммарный образ?
- как его использовать при распознавании?
- какое время потребуется для проведения необходимых вычислений?

Некоторые исследования этих вопросов даются в настоящей статье.

1. Размер сдвига, необходимый для совмещения образов

Оценка сходства двух образов проводится для нескольких взаимных положений образов, т.е. помимо центрированного положения исследуются также сдвиги образа в разных направлениях. В работе [3] был теоретически исследован вопрос о возможных ограничениях размера сдвигов. Изображение в [3] рассматривалось как пара, состоящая из множества I (ненулевые точки изображения), и плотности $f: I \rightarrow \mathbf{R}$ (распределения весов точек изображения). Степень совпадения эталонного и тестового изображений с плотностями f и g соответственно определялась как

$$\int |\varphi - g| d\sigma,$$

где мера $d\sigma$ определена в пространстве Ω ($I \subset \Omega$).

Рассматривалась задача поиска параллельного переноса эталонного образа, при котором его совпадение с тестируемым образом максимально. Было показано, что если наугад взятое совмещение удачно, то есть, если несовпадение эталонного и тестируемого образов составляет малую часть эталонного образа, то параллельный перенос, максимизирующий совпадение образов по длине составляет малую часть диаметра эталонного образа (диаметра выпуклой оболочки множества I).

В работе [3] было приведено доказательство следующего утверждения.

Для минимизации интеграла $\psi(a) = \int_{\mathbb{R}^n} |\varphi(x-a) - g(x)| d\sigma$ на прямой $a = |a|v$, натянутой на вектор v , достаточно рассматривать лишь векторы a , для которых выполнено неравенство

$$|a| < s_v = \frac{2\beta}{\alpha} d_v(\theta).$$

При $|a| > 2\beta d_v(\theta)/\alpha$ справедливо неравенство

$$\int_{\mathbb{R}^n} |\varphi(x-a) - g(x)| d\sigma \geq \beta,$$

$$\text{где } \alpha = \int_{\mathbb{R}^n} \varphi(x) d\sigma, \quad \beta = \int_{\mathbb{R}^n} |\varphi(x) - g(x)| d\sigma, \quad d_v(\theta) -$$

v -диаметр области Θ , Θ - выпуклое множество, вне которого $\varphi(x)=0$.

Если $\alpha \gg \beta$, то число $\beta / (\alpha - \beta)$ мало. Поэтому минимум $\Psi(a)$ достигается при длине вектора a , значительно меньшей, чем диаметр $d_v(\Theta)$ фигуры Θ в направлении вектора v .

Отсюда следует, что для достижения оптимального совмещения двух образов достаточно малых сдвигов в том случае, когда мера несовпадения при них незначительна.

2. Получение суммарного образа

Очевидный способ получения суммарного образа состоит в следующем. Выберем некоторый произвольный базовый образ объекта A_1 . Каждый из оставшихся образов ($A_2 \dots A_n$) модифицируем (путем сдвига) на пиксельной сетке так, чтобы расстояние $\mu_0(A_1, A_k)$ ($2 \leq k \leq n$) было минимальным. После этого для всех образов A_k ($1 \leq k \leq n$) просуммируем значения элементов a_{ij} в элементы матрицы

$$\Sigma(A_1, \dots, A_n) = \|\sigma_{ij}\|, \quad \sigma_{ij} = a_{ij1} + \sum_{k=2}^n a_{ijk}.$$

Понятно, что при таком способе действий результат суммирования зависит от выбора первого объекта A_1 (опорного образа). Рассмотрим эту зависимость на простейшем примере.

Будем считать, что есть сетка пикселей, каждый из которых имеет яркость от нуля до 1, причем 0 – соответствует белому цвету, а 1 – черному.

Пусть есть полоска высотой в один пиксель и шириной a ($1 \leq a \leq 5$). Если положить ее на сетку с ячейкой 1×1 так, чтобы горизонтальная граница совпадала с границей сетки, получим одно из положений, показанных на Рис.1.

Если считать, что полоска имеет чистый черный цвет, а фон – чисто белый, полоска займет три пикселя с яркостями $y, 1, a-y-1$ в первом случае ($y > a-1$) и два пикселя с яркостями y и $a-y$ во втором.

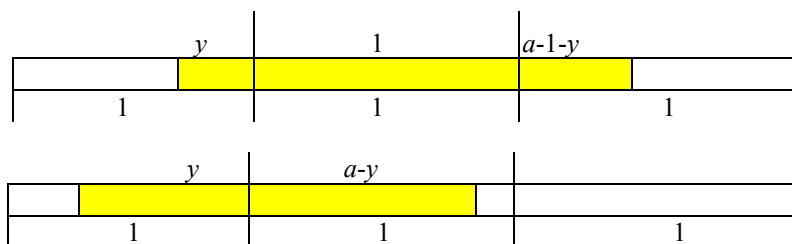


Рис.1. Примеры размещения полоски шириной a на сетке

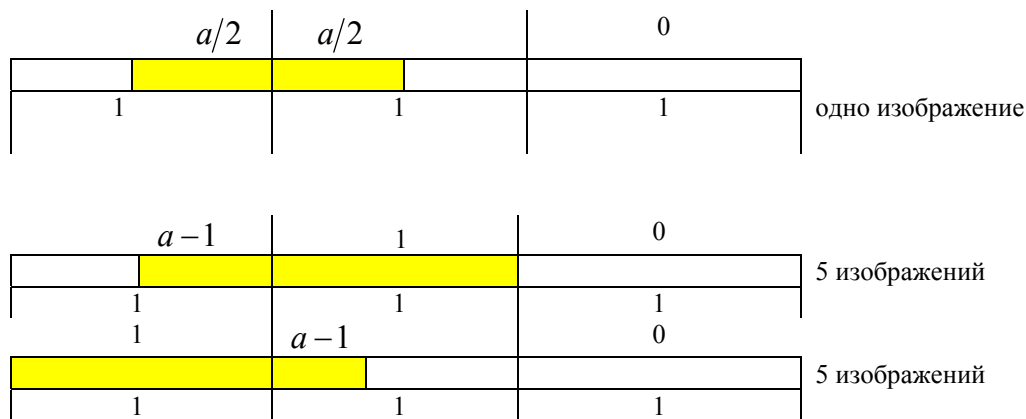


Рис. 2. Пример попадания полосы на сетку

Предположим, что у нас есть 11 изображений, в которых полоска попала на пиксельную сетку как показано на Рис.2. Если взять в качестве базового первое изображение, то изображения второго и третьего типа совместятся с ним наилучшим образом без сдвига, и при суммировании получается вектор

$$s_1 = (5.5a, 5.5a, 0),$$

Если же взять в качестве базового изображение второго типа, то изображения третьего типа совместятся с ним наилучшим образом, сдвинувшись вправо. Тогда после суммирования получим

$$s_2 = (5.5a - 5, 0, 5a + 10, 5a - 5).$$

$$\text{При } a = 1.1 \quad s_1 = (6.05, 6.05, 0),$$

$$s_2 = (1.05, 10.55, 0.5),$$

или после нормировки на число элементов

$$\bar{s}_1 = (0.55, 0.55, 0), \quad \bar{s}_2 = (0.1, 0.96, 0.05).$$

Видно, что влияние выбора опорного элемента на получающийся суммарный образ довольно велико. Можно попытаться его уменьшить или оптимизировать суммарный образ, например, перебрав опорные элементы и выбрав из них наилучший. Однако попробуем поставить задачу более четко, например, так.

Построим матрицу $B = \{b_{i,j}\}$ такую, чтобы сумма ее расстояний от объектов $\{A_k\}$ ($1 \leq k \leq n$) была минимальна. Проведем исследование на том же простейшем примере.

Зададимся сеткой, представляющей собою эталон из трех идущих подряд пикселей. В каждом записан некоторый вес – «яркость». Веса, записанные в этих трех пикселях, образуют вектор

$$X = (x_1, x_2, x_3), \quad 0 \leq x_1, x_2, x_3 \leq 1.$$

Вектор X будем рассматривать как некоторое полутоновое изображение. Рассмотрим размещенную на этой сетке полоску высотой единица и шириной a ($1 \leq a \leq 1.5$). Будем считать, что в левый пиксель попала ненулевая часть полоски шириной y . Полоска может пересекать либо два, либо три пикселя.

Таким образом, положение полоски также описывается вектором из трех компонент и однозначно определяется значением y :

$$Y = (y, 1, a - y - 1) \text{ если } y \leq a - 1 \text{ или}$$

$$Y = (y, a - y, 0) \text{ если } y > a - 1.$$

Теперь будем считать, что y есть случайная величина, равномерно распределенная на отрезке $[0,1]$, и посчитаем среднее значение расстояния до эталона X .

Наша цель - найти такой эталон X , при котором среднее значение расстояния полоски до эталона было бы минимальным. Из соображений симметрии ясно, что в таком «оптимальном» эталоне должно быть $x_1 = x_3$, и мы будем рассматривать только такие эталоны. Кроме того, очевидно должно быть $x_2 > x_1$.

Расстояние между эталоном и образом полоски равно:

$$d_2(X, Y) = \begin{cases} |x_1 - y| + (1 - x_2) + |x_1 - (a - y - 1)| & y \leq a - 1 \\ \min(|x_1 - y| + |x_2 - (a - y)| + x_1, x_1 + |x_2 - y| + |x_1 - (a - y)|) & y > a - 1 \end{cases} \quad (2)$$

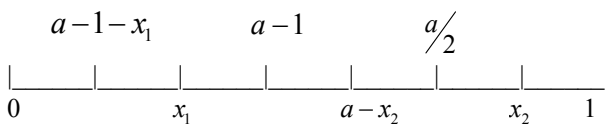
В случае $y > a - 1$ переход от $|x_1 - y| + |x_2 - (a - y)| + x_1$ к $x_1 + |x_2 - y| + |x_1 - (a - y)|$ соответствует возможному сдвигу полосы на 1, когда полоска из первых двух пикселей перемещается на два последних пикселя. Очевидно, что переход от первого случая ко второму происходит в точке $y = a/2$.

Среднее значение расстояния $M(X, Y)$ определяется интегралом

$$M(X) = \int_0^1 d_2(X, Y(y)) dy$$

В этом интеграле $d_2(X, Y)$ является линейной функцией на любом отрезке, где не меняет знака ни одно из выражений, входящих в $d_2(X, Y)$ под знаком модуля, и не происходит смены выражения, дающего минимум в формуле (2). Очевидно, задавшись взаимным расположением на оси некоторых простейших линейных функций от x_1, x_2, a можно вычислить $M(X)$ и найти x_1, x_2 , при которых достигается минимум.

Требуется проверить, соответствуют ли получившиеся значения x_1, x_2 сделанным предположениям. Приведем согласованное расположение критических точек, в которых может произойти излом подинтегральной кусочно-линейной функции:



Вычисленное в таких предположениях расстояние $M(X)$ равно

$$\begin{aligned} M(X) &= \int_0^{a-1} d_2(X, Y(y)) dy + \\ &+ \int_{a-1}^{a/2} d_2(X, Y(y)) dy + \int_{a/2}^1 d_2(X, Y(y)) dy = \\ &= \int_0^{a-1} (|x_1 - y| + 1 - x_2 + |x_1 - a + y + 1|) dy + \\ &+ \int_{a-1}^{a/2} (|x_1 - y| + |x_2 - a + y| + x_3) dy + \\ &+ \int_{a/2}^1 (x_1 + |x_2 - y| + |x_1 - a + y|) dy \end{aligned}$$

После взятия интегралов получим

$$M(X) = 2x_1^2 - 2(a-1)x_1 + 2x_2^2 - (2a+1)x_2 + 0.5a^2 + a$$

минимум которого

$$\min M(X) = -0.5a^2 + 1.5a - 5/8$$

достигается при значениях

$$x_1 = \frac{a-1}{2}, \quad x_2 = \frac{2a+1}{4},$$

т.е. оптимальный эталонный вектор имеет вид:

$$\left(\frac{a-1}{2}, \frac{2a+1}{4}, \frac{a-1}{2} \right).$$

Отметим, что x_2 при $1 \leq a < 1.5$ меньше единицы и достигает ее лишь при $x = 1.5$. Отсюда, в частности, следует, что наилучший эталонный вектор имеет сумму весов, меньшую a для $a < 1.5$. Такой вектор нельзя получить усреднением никаких экспериментально набросанных «полосок», так как сумма весов после нормировки во всех случаях будет равна a .

3. Использование суммарного образа

Из предыдущего рассмотрения видно, что непосредственное вычисление расстояния от нормированного суммарного образа до распознаваемого объекта не обещает надежных результатов. Кроме того, сравнение полутоновых изображений является операцией, требующей при исполнении больших затрат времени. Например, сравнение с помощью метрики μ_0 полутоновых и бинарных образов с размерами, кратным числу 64, различается на порядок. Так для полутоновых образов для операций вычитания и суммирования потребуется 4-кратное вычисление нескольких 128-битных инструкций SSE2, а для бинарных образов – одна единственная инструкция POPCNT, непосредственно реализующая метрику Хэмминга для бинарного образа из 64 точек. Поэтому далее рассмотрим методы сравнения, основанные на работе с бинарными образами.

Прежде всего, определим область, в которой вес любого элемента нормированного

суммарного образа Σ превышает некоторую константу ρ :

$$A_c = \{ (i, j) \mid 0 < i < m, 0 < j \leq n, \sigma_{i,j} \geq \rho \}.$$

Эту область A_c назовем общей областью. Константа ρ при полутоновых изображениях зависит от среднего значения «черных» точек, а в случае черно-белых равна 1.

Также определим слой точек L_1 , являющийся множеством точек, удаленных на 1 от общей области:

$$L_1 = \{ (i, j) \mid \sigma_{ij} > 0, [d_3((i, j), A_c)] = 1 \},$$

где $d_3(x, Y)$ - минимальное расстояние (например, Манхэттенское) от точки x до каждой из точек множества Y .

Все остальные точки, которые не входят в $A_c \cup L_1$ и значение которых превышает уровень фона, назовем граничными A_M . Сумму $s = |A_c| + |L_1| + |A_M|$ назовем площадью суммарного образа. Для оценки оптимальности полученной суммы мы будем рассматривать распределение мощностей $|A_c|$, $|L_1|$, $|A_M|$.

Пример общей области и границы приведен на Рис.3, на котором отображено распределе-

ние значений точек a_{ij} для множества просуммированных различных образов символа «н» одного шрифта. Этот суммарный образ характеризуется четко выделенной общей областью, сходной с образом «н», и границей, полученной из-за эффектов оцифровки.

Общая область может быть использована для определения функции близости бинарного и суммарного образов, отличающейся от метрики Хэмминга и вычисляемой как $d_1(B, A_c \cup L_1)$, где B - бинарный образ, а A_c - общая область суммарного образа $\Sigma(A_1, \dots, A_n)$. Эта псевдометрика, основанная на попадании одного образа в окрестность другого, существенно меньше зависит от эффектов оцифровки и имеет некоторое преимущество перед метрикой Хэмминга при кластеризации и распознавании. Для распознавания использовалась модифицированная функция близости на основе d_1 .

Рассмотрим два множества:

- C_0 - множество точек, принадлежащих A_c и отсутствующих в B ;
- C_1 - множество точек, принадлежащих B , и отсутствующих в $A_c \cup L_1$.

C_1 представим в виде двух подмножеств C_{11} , состоящего из точек, расстояние от которых до

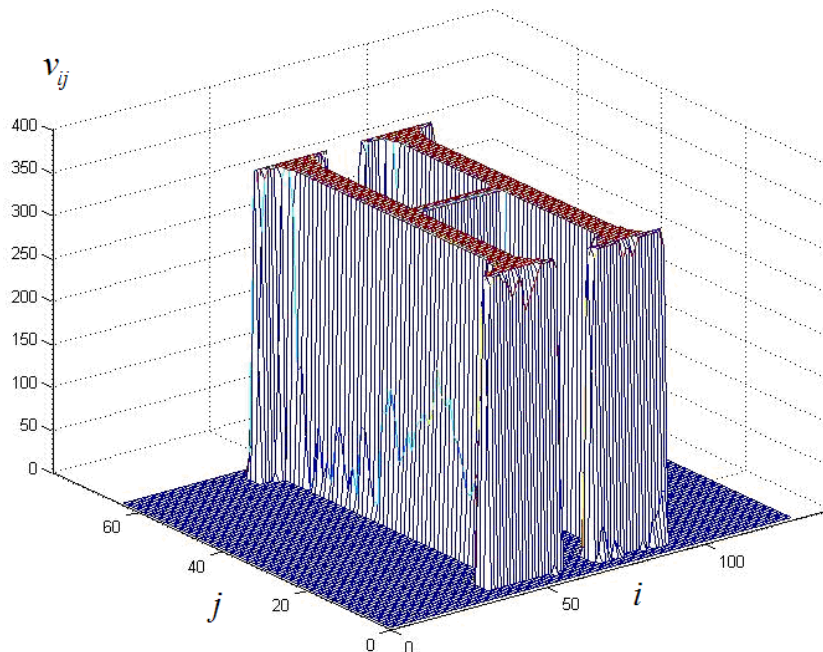


Рис. 3. Суммарный образ для символа «н»

$A_c \cup L_1$ не превышает 1, и подмножества C_{12} , состоящего из точек, расстояние от которых до $A_c \cup L_1$ превышает 1.

Близость образа B и суммарного образа $\delta_3(B, \Sigma(A_1, \dots, A_n))$ определим как $|C_0| + |C_{11}| + 2|C_{12}|$. Такое вычисление, как и раньше, проведем для нескольких взаимных положений образа B и суммарного образа $\Sigma(A_1, \dots, A_n)$, после чего выберем минимальное значение и возьмем его в качестве функции близости $d_3(B, \Sigma) = \min_{(h,v)} (\delta_3(B^{(h,v)}, \Sigma(A_1, \dots, A_n)))$.

4. Оценки быстродействия

Необходимо отметить, что вычисление расстояния d_1 непосредственно по формуле (1) требует слишком много операций. Для того чтобы проверить все точки образов, надо проверить каждую точку: черная она или белая, то есть принадлежит точке образу символа или нет. В первом случае необходимо проверить до восьми ее соседей.

Метрика Хэмминга μ_0 также может быть использована для поиска оптимального положения образа при его сравнении с опорным образом. Один из способов ее применения состоит в поиске оптимального положения каждого из образов по отношению к опорному образу, при этом производится несколько сдвигов так, чтобы расстояние $d_0(R^{(h,v)}, R_0)$ между образами этих элементами было наименьшим. Значение каждого из сдвигов h и v ограничено числом $n_0=2$.

Для построения другого алгоритма сравнения, основанного на метрике Хэмминга μ_0 , будем использовать дискретное преобразование Фурье (ДПФ). Пусть образы двух сравниваемых образов F и G задаются функциями $f(x,y)$ и $g(x,y)$, определенными на области прямоугольника с размерами $[1,m] \times [1,n]$.

Расстояние между образами F и G вычисляется с помощью метрики

$$\mu_4(F, G) = \iint_{\Omega} |f(x,y) - g(x,y)| dx dy.$$

Очевидно, что для образов μ_4 совпадает с рассмотренной выше метрикой Хэмминга μ_1 .

Аналогично d_0 и d_1 определим функцию близости

$$d_4(F, G) = \min(\mu_4(F^{(h,v)}, G),$$

учитывающие сдвиги, при этом ограничения на значения h и v не накладываются.

Иными словами, вычисление функции близости d_4 производится с помощью минимизации

$$\iint_{\Omega} |f(x,y) - g(x-h, y-v)| dx dy \rightarrow \min, \quad (3)$$

где (h,v) – вектор переноса образа G относительно образа F .

Минимизация (2) эквивалентна максимизации следующей функции:

$$f^*(h,v) = \iint_{\Omega} |f(x,y) \cdot g(x-h, y-v)| dx dy, \quad (4)$$

являющейся псевдосверткой функций $f(x,y)$ и $g(x,y)$. В (4) используется только сдвиг, а отражение, присущее стандартному виду свертки $\iint_{\Omega} |f(x,y) \cdot g(h-x, v-y)| dx dy$, отсутствует.

Аналогично известной теореме о свертке для псевдосвертки справедлива следующая зависимость:

$$\Phi(f^*(h,v)) = \mathcal{F}(f(h,v)) \cdot \overline{\Phi}(g(h,v)), \quad (5)$$

где $\mathcal{F}$ – преобразование Фурье, а функция $\overline{\Phi}(g(h,v))$ является комплексно-сопряженной функции $\Phi(g(h,v))$. С помощью дискретного преобразования Фурье из (5) можно найти значения аргументов $h_{\max}$ и $v_{\max}$, соответствующие максимальному значению функции $f^*(h,v)$. При подстановке в формулу (3) эти значения аргументов дадут минимальное значение расстояния d_2 между образами F и G , заданными с помощью функций $f(x,y)$ и $g(x,y)$ для всех возможных положений двух образов.

Вычисление дискретного преобразования Фурье возможно быстрее, чем перебор всех возможных векторов (h,v) для получения расстояния d_0 . Однако результаты раздела 1 дают нам возможность ограничить размеры вектора (h,v) , например, приняв $-2 \leq h, v \leq 2$. Если при этом $d_0(A, B)$ остается больше некоторого допустимого значения, образы A и B просто считаются различными. В противном случае $d_0(A, B)$ принимается за истинное расстояние.

В следующем параграфе будут рассмотрены результаты некоторых экспериментов, показывающих сравнительные скорости операций и оценки потерь от эвристик.

5. Эксперименты

Для вычисления $\mathcal{F}$ и $\mathcal{F}^{-1}$ использовалась библиотека `fftw-3.2.2` [4], предоставляющая возможности быстрого вычисления коэффициентов ДПФ. Теоретическое время работы алгоритма ДПФ, реализованного в библиотеке `fftw-3.2.2`, для образа размерности mn составляет $O(mn \ln(mn))$.

5.1. Получение суммарных образов различными способами

Был проведен эксперимент с несколькими множествами, содержащими образы, относящиеся к одному прообразу. Рассматривались множества M_1 и M_2 образов, полученных из одного идеального прообраза с помощью изменения масштаба и случайных искажений, а также множества отсканированных образов M_3 и M_4 (газета со средним качеством). Используемые множества описаны в Табл. 1, в которой для каждого множества приведены: его мощность, код символа-прообраза, шрифт, средняя высота и средний наклон образов.

Для каждого из множеств M производился поиск оптимального положения всех образов

по отношению к опорному образу с помощью способа сдвигов и с помощью преобразования Фурье. После суммирования образов с оптимальным взаимным расположением для обоих способов были получены визуально сходные суммарные образы, несмотря на то, что участвующие в суммировании отдельные бинарные образы значительно различались (Табл.2).

Суммарные образы анализировались с помощью распределения точек в общей области, первом слое и граничной области

$$|A_C|, |L_1|, |A_M|,$$

получившиеся распределения приведены в Табл.3 и подтверждают сходство полученных суммарных образов.

Анализ данных Табл. 3 и результатов аналогичных экспериментов свидетельствует о следующих закономерностях: площади общих областей, подсчитанные способами сдвигов с помощью d_0 и d_1 , могут сильно различаться (в два и более раз), однако при этом области $A_C \cup L_1$, подсчитанные этими же способами, различаются по площади менее значительно (не более чем на 20 %).

Табл. 1. Используемые в экспериментах множества образов

Мощность множества	Прообраз			
	Символ	Гарнитура	Высота	Угол наклона, град
M_1 (127)	A	Arial	16	0
M_2 (36)	D	Times New Roman	27	5
M_3 (127)	я	Times New Roman	20	6
M_4 (127)	я	Arial	20	3

Табл. 2. Примеры исходных бинарных и получившихся суммарных образов

	Примеры бинарных образов				Суммарные образы	
					Способ сдвигов	ДПФ
M_3						
M_4						

Табл. 3. Оценки распределения точек по слоям

Множество	Распределение точек $ A_C $; $ L_1 $; $ A_M $								
	Способ сдвигов d_0			Способ сдвигов d_1			ДПФ d_4		
	$ A_C $	$ L_1 $	$ A_M $	$ A_C $	$ L_1 $	$ A_M $	$ A_C $	$ L_1 $	$ A_M $
M_1	72	79	0	77	64	1	72	79	0
M_2	153	155	16	123	147	108	157	149	12
M_3	86	118	128	78	115	135	88	123	123
M_4	138	117	60	122	116	75	138	117	60

Табл.4. Оценки затрат времени на суммирование

Множество	Затраты времени на поиск оптимального взаимного положения (10^{-6} сек)		
	Способ сдвигов d_0	Способ сдвигов d_1	ДПФ d_4
M_1	140	152	9250
M_2	66	158	3235
M_3	181	452	9187
M_4	148	181	7731

5.2 Оценка быстродействия при суммировании образов различными способами

Для сравнения быстродействия работы различных способов суммирования оценивалось время, затраченное на поиск оптимального положения и вычисление суммы при оценке расстояния с помощью функций близости d_0 , d_1 и d_4 для нормализованных образов с размерами $m=n=64$. Эксперименты производились на компьютере CPU Intel Core 2 Quad Q9400, 2666 МГц, FSB=1333 МГц, с оперативной памятью DDR2 3 ГБ, DRAM Frequency=400 МГц, ОС MS Windows XP SP3. Как уже отмечалось, для вычисления БПФ при реализации функции d_4 использовалась библиотека fftw. Функции d_0 и d_1 были реализованы (запрограммированы в среде Microsoft Visual Studio 10.0) самостоятельно.

Результаты проведенных экспериментов приведены в Табл.4.

Вычислительные эксперименты показывают, что затраты времени на поиск оптимального взаимного положения для способов сдвигов, основанных на функциях близости d_0 и d_1 , либо близки, либо различаются в 2 – 2,5 раза. Способ

поиска оптимального взаимного положения для способа ДПФ по сравнению со способом сдвигов, основанным на функции близости d_0 , является существенно более медленным. Замедление на различных множествах составляет от 20 до 80 раз. Это вполне предсказуемо, так как алгоритм ДПФ является существенно более сложным, чем вычисление d_0 , оптимизированное описанными выше приемами.

Поясним для чего была рассмотрена функция d_1 , если при суммировании с использованием d_1 граница в полученном суммарном образе больше, чем для аналогичного суммирования с использованием d_0 и d_4 , а затраты времени на суммирование для d_1 выше чем для d_0 .

Рассмотрим алгоритм сравнения каждого из просуммированных образов $A_1, \dots, A_n$ с получившимся различными способами суммарным образом $\Sigma(A_1, \dots, A_n)$ (для функции близости d_0 , для d_1 и для d_3). Для этого используем функцию близости d_3 , а для оценки рассмотрим средний штраф

$$\sum_{k=1}^n d_3(A_k, \Sigma(A_1, \dots, A_n)) / n. \quad (6)$$

Табл. 5. Средний штраф по формуле (6) для различных способов суммирования

Множество	Средний штраф по формуле (6)		
	Способ сдвигов d_0	Способ сдвигов d_1	ДПФ d_4
M_1	1.362205	1.803150	1.362205
M_2	7.166667	3.638889	5.583333
M_3	5.708661	5.039370	5.929134
M_4	3.911765	3.735294	3.911765

Результаты сведены в Табл. 5 и свидетельствуют о наилучшем балансе способа суммирования с помощью способа сдвигов d_1 и оценки близости образа и суммарного образа с помощью d_3 . В реальности, возможны случаи, когда средний штраф для d_1 оказывается выше, чем для d_0 и d_4 , однако для отсканированных образов применение d_1 для распознавания дает лучшую точность.

Заключение

Из предыдущего рассмотрения и результатов экспериментов можно сделать некоторые выводы.

1. Ограничения на сдвиги при вычислении расстояния d_0 в виде $-2 \leq h, v \leq 2$ являются достаточно надежной эвристикой. Отклонения в количестве точек в области $A_c \cup L_1$ незначительны.

2. Точное вычисление сдвигов при помощи ДПФ требует значительных вычислительных затрат, хотя использование полученных суммарных образов позволяет при распознавании получить более точные результаты.

3. Использование суммарного образа $\Sigma(A_1, \dots, A_n)$ для распознавания путем непосред-

ственного вычисления расстояния $d_0(B, \Sigma(A_1, \dots, A_n))$ не обеспечивает оптимальных результатов при распознавании.

4. В случае использования функции расстояния $d_1(B, \Sigma(A_1, \dots, A_n))$ на основе псевдометрики μ_1 увеличивается площадь суммарного образа, а получение суммарного образа при помощи функции расстояния d_1 требует больших вычислительных затрат, но при этом получаются более точные результаты при распознавании.

Литература

1. LeCun Y., Cortes C., Burges C.J.C. The MNIST Database of handwritten digits [Электронный ресурс]. – Режим доступа <http://yann.lecun.com/exdb/mnist/index.html>. Дата обращения: 16.10.2014.
2. Котович Н.В. Алгоритмы кластеризации образов символов // Труды ИСА РАН "Обработка изображений и анализ данных". — М.: Эдиториал УРСС, 2008, Т.38. - С. 241-251.
3. Арлазаров В.Л., Славин О.А., Фарсобина В.В., Хованский А.Г. Поиск оптимального положения при сравнении оцифрованных образов символов // Искусственный интеллект и принятие решений, Т.3. М.: ПолипринтСервис, 2013. С. 48-59.
4. FFTW [Электронный ресурс]. – Режим доступа <http://www.fftw.org/>. Дата обращения: 16.10.2014.

Арлазаров Владимир Львович. Заведующий лабораторией, профессор Института системного анализа РАН. Окончил МГУ им. М.В. Ломоносова в 1961 году. Член-корреспондент РАН. Автор более 100 печатных работ. Область научных интересов: теория графов, распознавания образов, программирование. E-mail: Arl@mail.ru.

Славин Олег Анатольевич. Заведующий лабораторией Института системного анализа РАН. Окончил Московский институт радиотехники, электроники и автоматики в 1988 году. Доктор технических наук. Автор 69 печатных работ и одной монографии. Область научных интересов: распознавания образов, информационные системы. E-mail: OlegSlavin@mail.ru.

Фарсобина Вера Викторовна. Научный сотрудник института системного анализа РАН. Окончила Московский экономико-статистический институт в 1979 году. Автор 23 печатных работ. Область научных интересов: распознавания образов, вычислительные эксперименты. E-mail: vera@cs.isa.ru.