

О восстановлении идеального прообраза по коллекции образов

Ю. В. Титов

В статье рассмотрены некоторые аспекты одного алгоритма восстановления идеального прообраза, использующего набор многократно отсканированных изображений одного символа-оригинала.

1. Введение

В современных алгоритмах распознавания символов основанных на двухпроходной схеме одним из основных разделов второго прохода процесса распознавания является кластеризация символов, уже распознанных на первом проходе [1].

Кластеризация представляет собой разделение множества образов распознанных как одноименные символы на несколько частей внутри своей группы. По результатам кластеризации, при использовании дополнительных методов, можно выделить атрибуты символов (жирность, курсив), определить многообразие шрифтов в распознаваемой области, обнаружить ошибки первого прохода распознавания, в том числе ошибки сегментации символов внутри строки.

Для кластеризации необходимо использование какой-либо меры различия двух образов. Среди наиболее часто используемых мер можно назвать интегральную меру наложения, модернизированную меру Хаусдорфа, упрощенную интегральную меру и т. д.

Все образы полученного кластера складываются в стопку друг на друга, и получается некоторое обобщение всех символов кластера. Данную конструкцию называют *взвешенным растром*. Собранный взвешенный растр можно использовать для получения оценок принадлежности к данному кластеру отдельных его элементов, или для сравнения кластеров между собой.

Очевидно, что от используемой меры напрямую зависит результат, — какие кластеры получатся. Однако, это не единственный параметр, если

смотреть дальше, и задаться вопросом — какие именно взвешенные растры получатся? В данной статье будут подробнее рассмотрены описанные процессы без углубления в алгоритмы кластеризации, наибольший акцент будет сделан на анализ взвешенного растра. В частности будет предложен метод оценки и алгоритм для сравнения плотности упаковки растров одного кластера.

2. Постановка задачи

Источником бинарных растров для нас является хранилище в виде базы графических образов [2]. Рассмотрим растры $\mathbf{R}(\mathbf{m}, \mathbf{n}) = \|\mathbf{r}_{ij}\|$, где \mathbf{m} (ширина) и \mathbf{n} (высота) — размеры растра, $\mathbf{i} = \overline{1, \mathbf{m}}$, $\mathbf{j} = \overline{1, \mathbf{n}}$, где \mathbf{r}_{ij} — значением растра в точке (\mathbf{i}, \mathbf{j}) , равное 0 или 1. Назовем *точкой растра* (или *пикселем*) $\mathbf{R}(\mathbf{m}, \mathbf{n}) = \|\mathbf{r}_{ij}\|$, где $\mathbf{i} = \overline{1, \mathbf{m}}$, $\mathbf{j} = \overline{1, \mathbf{n}}$, пару чисел (\mathbf{i}, \mathbf{j}) , где \mathbf{i} и \mathbf{j} — значения индексов элемента \mathbf{r}_{ij} . Будем обозначать ее $\mathbf{P}(\mathbf{i}, \mathbf{j})$. Назовем \mathbf{r}_{ij} *значением растра* в точке $\mathbf{P}(\mathbf{i}, \mathbf{j})$. Два растра $\mathbf{R}_1(\mathbf{m}_1, \mathbf{n}_1)$ и $\mathbf{R}_2(\mathbf{m}_2, \mathbf{n}_2)$ называются *равновеликими*, если $\mathbf{m}_1 = \mathbf{m}_2$ и $\mathbf{n}_1 = \mathbf{n}_2$. Пусть \mathbf{S} — произвольное множество равновеликих растров $\mathbf{S} = \{\mathbf{R}_k(\mathbf{m}, \mathbf{n})\} = \{\|\mathbf{r}_{ij}^{(k)}\|\}$, где $\mathbf{k} = \overline{1, \mathbf{N}}$, $\mathbf{i} = \overline{1, \mathbf{m}}$ и $\mathbf{j} = \overline{1, \mathbf{n}}$. Тогда *суммой* этих растров назовем растр $\mathbf{R}(\mathbf{m}, \mathbf{n}) = \|\mathbf{r}_{ij}\|$ такой, что

$$\mathbf{r}_{ij} = \sum_{\mathbf{k}=1}^{\mathbf{N}} \mathbf{r}_{ij}^{(\mathbf{k})}. \quad (1)$$

Тем самым взвешенный растр является суммой растров коллекции \mathbf{S} . Зададимся целью нахождения способа наиболее плотной укладки взвешенного растра, для этого в частности нужно предложить адекватную меру плотности, т. е. определить само понятие «плотность укладки». Собранный наиболее плотным образом взвешенный растр после бинаризации и будет представлять собой идеальный прообраз.

3. Простейший случай

При сканировании положение объекта на сетке сканера будем считать равномерно распределенной величиной. Подразумевается, что нам интересно не абсолютное расстояние от краев области сканирования, а положение относительно границ ближайших ячеек. То есть, попал ли объект на сетку сканера, аккуратно накрыв целое количество ячеек, или полоска

накрыла две ячейки (два пикселя) целиком, и еще у двух ячеек накрыла их половину.

Будем считать, что при бинаризации отдельный пиксель становится черным, если численное значение его оттенка, выраженное в стандартной шкале от 0 до 255, оказалось ближе к черному краю шкалы (0 — черный, 255 — белый). В противном случае — пиксель становится белым.

Будем говорить, что пиксель во взвешенном растре *темный*, если доля бинаризованных растров покрывающих его превосходит 0,5, т. е.

$$r_{ij} = \sum_{k=1}^N r_{ij}^{(k)} > \frac{N}{2}. \quad (2)$$

В противном случае такой пиксель будем называть *светлым*. Соответственно будем применять формулировки «становится темнее» и «становится светлее», если хотим показать, что после какого-либо преобразования доля покрывающих растров для темных участков увеличилась, а для светлых уменьшилась.

Рассмотрим вначале взвешенный растр, получающийся при сканировании и последующем складывании в стопку простейших объектов — вертикальных полосок, при этом будем следить лишь за одним из горизонтальных сечений, чтобы не отвлекаться на концы полосок. Для определенности возьмем горизонтальное сечение полоски с номером $n/2$.

Поскольку ширины растров-полосок могут получиться различными, а сумму растров мы определяли только для равновеликих растров, то будем считать, что для $\forall \mathbf{R}_k(\mathbf{m}_i, \mathbf{n}_j) \subset \mathbf{S} = \{\mathbf{R}_k(\mathbf{m}, \mathbf{n})\} = \{\|\mathbf{r}_{ij}^{(k)}\|\}$, где $\mathbf{i} = \overline{1, m}$, $\mathbf{j} = \overline{1, n}$, $\mathbf{k} = \overline{1, N}$ существует \mathbf{M}_{\max} и \mathbf{N}_{\max} , такие что $\forall \mathbf{m}_i < \mathbf{M}_{\max}$, $\forall \mathbf{n}_j < \mathbf{N}_{\max}$. Тогда каждый растр, участвующий в выборке на кластеризации, стандартизуем до размера \mathbf{M}_{\max} , \mathbf{N}_{\max} . Имеем в итоге стандартизованный набор растров $\mathbf{R}_1(\mathbf{m}_1 + \mathbf{x}_1, \mathbf{n}_1 + \mathbf{y}_1)$, $\mathbf{R}_2(\mathbf{m}_2 + \mathbf{x}_2, \mathbf{n}_2 + \mathbf{y}_2)$, ... $\mathbf{R}_k(\mathbf{m}_i + \mathbf{x}_k, \mathbf{n}_j + \mathbf{y}_k)$ такой, что $\mathbf{m}_i + \mathbf{x}_k = \mathbf{M}_{\max}$, $\mathbf{n}_j + \mathbf{y}_k = \mathbf{N}_{\max}$, где $\mathbf{i} = \overline{1, m}$, $\mathbf{j} = \overline{1, n}$, $\mathbf{k} = \overline{1, N}$. Вопрос позиционирования растра $\mathbf{R}_k(\mathbf{m}_i, \mathbf{n}_j)$ внутри большего растра $\mathbf{R}_k(\mathbf{m}_i + \mathbf{x}_k, \mathbf{n}_j + \mathbf{y}_k)$ рассмотрим чуть ниже. В общем случае возможно применение нормализации растра произвольного размера, т. е. сжатие к растру фиксированного размера с последующей бинаризацией [3].

Принимая во внимание равномерное распределение границ полосок на сетке сканера, можно показать, что средняя ширина d коллекции \mathbf{S} бинаризованных полосок будет совпадать с шириной полоски-оригинала.

$$d = \frac{1}{N} \sum_{k=1}^N \sum_{j=1}^m r_{n_j}^{(k)}.$$

К примеру, после бинаризации 20 полосок мы получили 10 полосок ширины в 1 пиксель, и 10 полосок ширины 2 пикселя. В этом случае, как утверждается, ширина оригинальной полоски, которую сканировали 20 раз, получая указанную коллекцию образов, составляет 1,5 пикселя. В данном случае мы пренебрегаем погрешностью сканера и появлением дополнительных искажений.

Спрашивается, каким образом лучше складывать нашу коллекцию S в стопку, чтобы получившийся взвешенный растр удовлетворял бы требованиям поставленной задачи, т. е. его укладка в стопку была бы наиболее плотной? Для начала посмотрим, что в принципе может получиться, и какой из возможных взвешенных растров $R(\mathbf{m}, \mathbf{n})$ стоит считать уложенным более плотно, а какой менее плотно.

В случае растров-полосок вариации получаются из-за различного позиционирования растров при стандартизации. Во-первых, можно прижимать растр $R_k(\mathbf{m}_i, \mathbf{n}_j)$ к одному — для определенности левому — краю растра $R_k(\mathbf{m}_i + \mathbf{x}_k, \mathbf{n}_j + \mathbf{y}_k)$, однако в этом случае нарушается симметричность взвешенного растра — он получается сильно темнее с левой стороны. Если мы хотим сохранить симметричность, то можно попытаться центрировать образы — второй вариант. Однако часто оказывается не возможно провести центрирование с точностью до пикселя — хотя бы в описанном выше примере с 20 полосками — без введения «дробного» сдвига погрешность центрирования будет составлять 1 пиксель, а при выборе в спорных случаях всегда, к примеру, левого положения мы будем иметь первый вариант. Если же выбирать приоритетное положение случайным образом, то получается третий вариант.

4. Мера плотности укладки

Для оценки плотности укладки необходимо ввести меру плотности для взвешенного растра. Идеальным вариантом, т. е. самой плотной укладкой является случай, при котором все растры совпадают (т. е. равновелики, и $r_{ij}^{(k)} = r_{ij}^{(p)}$ для произвольных k и p от 1 до N), и значение темных пикселей во взвешенном растре, т. е. величина «какая доля растров покрывает данный пиксель» равна единице. Пиксели другого типа отсутствуют вовсе.

Другой крайний случай — наименее плотная укладка — мог бы реализоваться, если бы каждый пиксель взвешенного растра покрывался только одним бинаризованным растром, т. е. в сумме (1) r_{ij} принимало бы значение только 0 или 1.

Наиболее естественным кажется вычисление средней поточечной плотности. То есть вычислять среднее арифметическое (по всем точкам

взвешенного растра) величины $\rho(i, j) = \text{«доля растров, накрывающая данный пиксель»}$.

$$M(R) = \frac{1}{Vr(R)} \sum_{i=1}^n \sum_{j=1}^m \rho(R, i, j),$$

где $\rho(R, i, j) = \frac{1}{N} \sum_{k=1}^N r_{ij}^{(k)}$ — плотность в точке с координатами (i, j) , а $Vr(R)$

— площадь взвешенного растра $R(\mathbf{m}, \mathbf{n})$, т. е. количество точек, значение которых отлично от нуля:

$$Vr(R) = \sum_{i=1}^n \sum_{j=1}^m I(r_{ij}),$$

где $I(r_{ij})$ — индикаторная функция. Значения $M(R)$ лежат в отрезке от 0 до 1, а для увеличения плотности взвешенного растра необходимо увеличивать $M(R)$.

В описанном примере с 20 полосками при первом варианте сборки растра плотность средняя получается $(1 + 0,5) / 2 = 0,75$. При третьем алгоритме укладки — тоже 0,75. При максимизации средней плотности, будет достигаться увеличение доли темных пикселей, и уменьшение доли светлых. Возможно увеличение средней плотности за счет уменьшения площади взвешенного растра, сохраняя при этом коллекцию S .

Видны недостатки этой меры — первый взвешенный растр интуитивно уложен более плотно, т. е. имеет больше темных участков, за счет осветления остальных пикселей. Так же при большом количестве светлых пикселей средняя плотность будет мала, в то время как взвешенный растр может быть вполне плотным (например, когда один образ из коллекции совсем плохо вкладывается в стопку, а остальные растры — близки друг другу по форме и размерам).

Рассмотрим взвешенный растр и его же бинаризацию $\text{Bin}(R)$. При бинаризации по медиане (среднему уровню) темные пиксели взвешенного растра становятся черными, а светлые — белыми. Возможен выбор порога бинаризации отличного от медианы. В этом случае в (2) в правой части неравенства выбирается значение отличное от $N/2$.

Введем плотность бинаризованого взвешенного растра. Пусть $\rho(\text{Bin}(R))$ равна единице для всех черных пикселей бинаризованого растра, и равна нулю для остальных пикселей. Предлагается взять в качестве альтернативной меры плотности укладки взвешенного растра пото-

чечную сумму модулей разности плотностей взвешенного растра и его бинаризации:

$$B(R) = \sum_{i=1}^n \sum_{j=1}^m |\rho(R, i, j) - \rho(\text{Bin}(R), i, j)|.$$

Заметим, что для увеличения плотности укладки взвешенного растра R мы стремимся уменьшить B -меру, в то время как среднюю плотность $M(R)$ мы старались увеличить (максимальное значение B -меры может быть сколь угодно большим).

B -мера для первого варианта укладки нашей 20-пиксельной коллекции равна 0,5. Для третьего варианта B -мера тоже принимает значение 0,5. То есть B -мера также не лишена первого описанного недостатка. Однако B -мера лучше средней плотности $M(R)$ позволяет отследить увеличение плотности укладки в тех случаях, когда во взвешенном растре присутствуют в большом количестве очень светлые пиксели. Средняя плотность при этом сильно уменьшается, в то время как B -мера почти не изменяется. К примеру, пусть коллекция из 20 образов состоит из одной полоски ширины 3, 18 полосок ширины 1 и еще одна полоска ширины 2. Средняя плотность примет значение $(0,05 + 0,1 + 1)/3 = 0,38$. B -мера в этом примере равна $(0,05-0) + (0,1-0) + (1-1) = 0,15$. Ввиду того, что мы отказались от нормировки по площади для B -меры, сравнительное применение её возможно только в пределах одной коллекции. Однако, для сравнения методов влияющих на плотность укладки взвешенного растра этого вполне достаточно.

Если нам потребуется ввести меру, которая бы могла оценить плотность укладки взвешенного растра при одинаковой средней плотности, то можно будет применить один из множества известных алгоритмов определения контрастности: средний градиент, методы, основанные на анализе гистограмм и т.п. Однако в практических задачах распознавания при сборке взвешенного растра такая точность не требуется, так как даже при незначительном увеличении плотности укладки, меры, предложенные выше, позволяют принять решение об оценке того или иного алгоритма.

5. Укладка сложных растров

Как было показано выше, для простейшего примера с полосками наиболее плотная укладка взвешенного растра получалась при простом сдвиге в процессе стандартизации всех полосок к одному краю. Для растров с более сложной формой такой простой прием перестает работать:

небольшие выступающие части могут сдвинуть растр внутри стопки на заметное расстояние, сильно ухудшив его положение в целом.

Наиболее естественным представляется при добавлении очередного растра выбирать такое его положение в стопке, чтобы сумма выступающих и недостающих частей относительно эталона (к примеру, первого растра) в стопке было бы минимально. Этот метод дает неплохой результат, однако имеется большая зависимость от случайного фактора — выбора эталона. Можно развивать этот прием, используя адаптирующийся эталон — при добавлении очередного растра в качестве эталона используется нормированная сумма уже добавленных растров.

В программе распознавания текстов Cognitive Forms используются некоторые комбинации описанных методов, в основу которых легла мера различия Хаусдорфа, а именно: расстоянием между двумя растрами считается сумма двух величин. Первая из них: на столько пикселей надо расширить первый растр, чтобы в него поместился второй, вторая величина — наоборот.

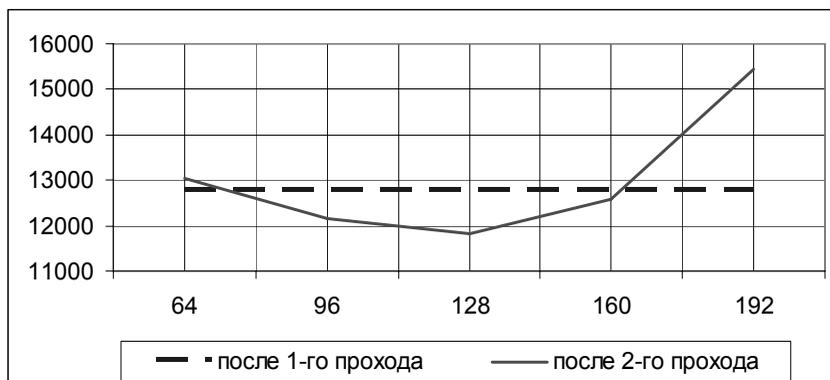
При такой метрике результат кучности взвешенного растра оказывается несколько лучше. В данной статье предлагается *метод второго прохода* при сборке взвешенного растра, позволяющий еще в большей степени увеличить плотность укладки и одновременно практически избавиться от сильного влияния случайных факторов. Для оценки улучшения кучности сборки взвешенного растра мы будем применять В-меру.

Суть нововведения заключается в использовании второго прохода при сборке растра. При первом проходе растры $\mathbf{R}_k(\mathbf{m}_k, \mathbf{n}_k) \subset \mathbf{S} = \{\mathbf{R}_k(\mathbf{m}, \mathbf{n})\}$ накладываются на первый из них $\mathbf{R}_1(\mathbf{m}_1, \mathbf{n}_1)$. После этого предлагается бинаризовать получившийся взвешенный растр, т. е. отбросить те пиксели, доля накрытия которых исходными растрами меньше заранее определенной доли (порога бинаризации). После этого делаем второй проход, т. е. используем получившуюся бинаризацию $\mathbf{Bin}(\mathbf{R})$ как эталон для накладки вариантов образов коллекции \mathbf{S} .

Приведем результаты значения В-меры для результатов первого и второго прохода. В тесте использовалась коллекция из 4 000 латинских букв F, напечатанных шрифтом Times New Roman, сканирование проводилось с разрешением 300 dpi. Шрифт и буква были выбраны специально, чтобы показать эффект второго прохода на весьма сложных объектах. Параллельно исследовалось оптимальное значение порога бинаризации для взвешенного растра первого прохода.

На рисунке по вертикальной оси отложено значение В-меры, по горизонтальной — пороговое значение бинаризации, применявшееся для получения эталона из взвешенного растра. Пунктиром отмечено значение В-меры после первого прохода. Как видно из графика, оптимальным является пороговое значение 128 (по шкале от 0 до 255), и его использование дает улучшение значения В-меры кучности взвешенного растра на 10 %.

Аналогичные результаты наблюдались и для других букв. Стоит отметить: чем проще геометрически оказывался символ, тем меньше оказывалось относительное уменьшение В-меры. Увеличение В-меры при пороге бинаризации равным 128 после 2-го прохода нигде зафиксировано не было.



Третий проход практически не вносил заметных изменений, и его применение не представляется актуальным.

Литература

1. Арлазаров В. Л., Котович Н. В., Славин О. А. Адаптивное распознавание // Информационные технологии и вычислительные системы. 2003. № 3.
2. Славин О. А. Средства управления базами графических образов символов и их место в системе распознавания // Сб. тр. ИСА РАН «Развитие безбумажных технологий в организациях». 1999. С. 317–330.
3. Славин О. А., Подрабинович А. А. Древоподобное распознавание нормализованных символов // Сб. тр. ИСА РАН «Интеллектуальные технологии ввода и обработки информации». 1998. С. 137–156.