

# Моделирование поведения и интеллекта

УДК 681.39

## ДВУМЕРНОЕ ПРОГРАММИРОВАНИЕ В ЗАДАЧАХ АНАЛИЗА ИЗОБРАЖЕНИЙ

В. К. КОВАЛЬ, М. И. ШЛЕЗИНГЕР

(Киев)

Описывается и исследуется алгоритм анализа зрительных изображений. Алгоритм основан на использовании специфических грамматик, порождающих изображения, а не последовательности, и названных двумерными грамматиками.

### 1. Специфика задач распознавания изображений

Распознавание зрительных сигналов является исключительно своеобразной составной частью распознавания образов.

В целом ряде задач распознавания незрительных сигналов, перечисленных, например, в монографии Вапника и Червоненкиса [1], основная трудность усматривается в том, что множества сигналов, соответствующих одному классу, заранее неизвестны. Трудности этого рода преодолеваются применением тех или иных алгоритмов обучения. Восстановление в процессе обучения необходимых множеств фактически завершает работу над решением задачи распознавания.

Своеобразие задач зрительного анализа заключается в том, что наличие сведений о множествах сигналов, соответствующих классам, не завершает решение задачи распознавания, а является лишь началом поисков, не всегда завершающихся успехом.

Рассмотрим простейший пример задачи распознавания, точное решение которой к настоящему времени неизвестно, несмотря на то, что классы распознаваемых сигналов в этой задаче полностью и однозначно определены.

*Пример 1.* Пусть известно, что распознаваемые изображения заданы на прямоугольном поле зрения размером  $m \times n$ . Это значит, что поле зрения представляет собой множество  $T$  пар  $(i, j)$ ,  $i=1, \dots, m$ ;  $j=1, \dots, n$ , а изображение представляет собой отображение множества  $T$  на алфавит  $V$  сигналов. Будем предполагать, что алфавит  $V$  состоит из двух сигналов,  $V = \{0, 1\}$ . Зададим некоторое множество, состоящее из  $m+n$  элементарных эталонных изображений, обозначенных через  $v_i^A$  ( $i=1, \dots, m$ ) и  $v_j^B$  ( $j=1, 2, \dots, n$ ). Изображение  $v_i^A$  называется горизонтальной линией в  $i$ -й строке, а  $v_j^B$  — вертикальной линией в  $j$ -м столбце.

Изображение  $v_i^A$  имеет вид:

$$v_i^A(i, j) = 1, \text{ если } i = i^*, j = 1, \dots, n;$$
$$v_i^A(i, j) = 0 \text{ в противном случае.}$$

Аналогично изображение  $v_j^B$  имеет вид:

$$v_j^B(i, j) = 1, \text{ если } j = j^*, i = 1, \dots, m;$$
$$v_j^B(i, j) = 0 \text{ в противном случае.}$$

Зададим операцию наложения изображений. Если изображение  $v$  является наложением изображений  $v$  и  $v'$ , то  $v(i, j) = 0$  в том случае, если и  $v'(i, j) = 0$ , и  $v''(i, j) = 0$ , а в противном случае  $v(i, j) = 1$ .

Зададим множество  $E$  эталонных изображений. В это множество входят: изображения  $v(i, j) = 0$  для всех  $i, j$ ; все элементарные эталонные изображения; а также любое изображение  $v$ , которое может быть представлено в виде наложения изображений, входящих в это множество. Очевидно, что множество  $E$  состоит из  $2^{n+m}$  различных эталонных изображений.

На вход распознающего устройства подается изображение  $x$ , являющееся результатом случайного искажения одного из эталонных изображений. Это значит, что если в эталонном изображении для некоторой пары  $i, j$   $v(i, j) = a$  ( $a = 1, 0$ ), то в искаженном изображении для этой пары  $x(i, j) = 1 - a$  с вероятностью  $p$  и  $x(i, j) = a$  с вероятностью  $1 - p$ . Распознающее устройство должно для любого искаженного изображения указать наиболее вероятное эталонное изображение или, что то же самое, указать изображение  $v$ , которое обеспечивает минимум выражения

$$\sum_{i,j} |x(i, j) - v(i, j)|.$$

Кажущаяся простота этой задачи является в высшей степени обманчивой. К настоящему времени неизвестен ни один алгоритм распознавания, более того, ни один алгоритм оптимизации, который можно было бы применить для успешного решения сформулированной задачи. При этом следует иметь в виду, что здесь говорится о точном решении, а не о наборе маленьких хитростей, которые, как часто пишут, «хотя и не гарантируют, но в большинстве практических случаев позволяют надеяться» и т. д.

Не умаляя значение разнообразных эвристических приемов, позволяющих решать многие практические задачи, к сформулированной задаче хотелось бы относиться с повышенной строгостью. Дело в том, что реальные задачи значительно сложнее приведенного иллюстративного примера. Теория распознавания должна указывать пути создания алгоритмов для анализа аэрофотоснимков, хромосом, треков элементарных частиц и т. п. В каждом из этих случаев требуется обнаруживать объекты весьма замысловатой конфигурации на фоне и в окружении не менее сложных объектов. Естественно, что в такой простейшей задаче, как сформулированная выше, должна достигаться полная ясность при отсутствии каких-либо дополнительных упрощающих предположений.

Приведенный пример иллюстрирует основные трудности решения каждой отдельно взятой задачи распознавания, находящиеся вне сферы теории обучения. А именно, основная трудность заключается в том, что для распознавания каждого отдельно взятого изображения требуется оптимизировать функцию, заданную на множестве, которое весьма необычно для известных, хорошо изученных методов оптимизации. Объектом теории распознавания, таким образом, должно быть изучение свойств этих специфических множеств, разработка формализма, позволяющего задавать разнообразные множества, уместные при решении задачи распознавания изображений. Этот формализм должен обладать универсальностью, т. е. должна существовать возможность задания любого множества на языке этого формализма. Кроме того, этот формальный язык должен быть удобен с точки зрения задач распознавания изображений. Это значит, что множества, которые уместно назвать зрительным образом, должны задаваться на этом формальном языке с помощью сравнительно короткого предложения. И, наконец, важнейшее требование, формальный язык должен быть конструктивным, т. е. для любого множества, просто записываемого на данном языке, должен существовать простой алгоритм оптимизации функции на этом множестве. Иными словами говоря, должен существо-

вать простой транслятор, который для каждой формальной записи задачи распознавания указывал бы алгоритм решения этой задачи. Описание такого формализма посвящен следующий раздел.

Заканчивая данное введение, необходимо отметить, что предлагаемый нами формализм служит для задания конструкций, подобных формально-грамматическим, но является двумерным обобщением последних. Ряд конструкций такого типа, названных двумерными грамматиками, описан в иностранной литературе [2—5]. Наше стремление дополнить этот ряд еще одной двумерной грамматикой объясняется тем, что ни одна из известных двумерных грамматик не удовлетворяет указанному выше требованию конструктивности. Это значит, что представление какого-либо множества изображений в виде одной из известных грамматик не облегчает и не указывает путей решения задачи распознавания, т. е. решения задачи оптимизации на множестве порождаемых грамматикой изображений. Можно надеяться, что предлагаемая нами двумерная грамматика именно этим недостатком не обладает.

По своему характеру данная работа наиболее близка к циклу работ Ковалевского, в которых рассматриваются алгоритмы анализа сложных изображений с применением динамического программирования [6]. Существенным же отличием данной работы от работ указанного цикла является рассмотрение именно двумерных, а не одномерных грамматик.

## 2. Двумерные грамматики и двумерное программирование. Основные понятия

Конечное множество  $Tv$  будет называться полем зрения, множество  $V$  — алфавитом сигналов, а функции  $v: Tv \rightarrow V$  — изображением.

Конечное множество  $Ts$  будет называться полем описания, множество  $S$  — алфавитом структурных элементов, а функция  $s: Ts \rightarrow S$  — описанием.

Объединение поля зрения и поля описания будет называться полем и обозначаться  $T$ , объединение алфавитов  $V$  и  $S$  будет обозначаться  $Z$ . Буквы  $v, s, t, z$  будут обозначаться элементы множеств  $V, S, T, Z$  соответственно.

Элемент множества  $T$  будет называться клеткой, а элемент алфавита  $Z$  — символом.

Вариантом  $z$  будет называться пара  $(v, s)$ , т. е. функция  $T \rightarrow Z$ , такая, что  $z(t) \in V$ , если  $t \in Tv$ , и  $z(t) \in S$ , если  $t \in Ts$ .

Две клетки  $t$  и  $t'$ , принадлежащие полю, будут называться соседними, если некоторый симметричный предикат  $R(t, t')$  равен 1. Предполагается всюду в данной работе, что  $R(t, t) = 0$  для всех  $t \in T$ , т. е. клетка не является соседней с собой. Множество пар соседних клеток обозначим через  $N$ . Для любой пары  $(t, t')$  соседних клеток задано множество  $ZZ(t, t')$  допустимых пар вида  $(z, z')$ .

Вариант  $z$  называется допустимым, если для любой пары  $(t, t') \in N$  справедливо  $(z(t), z(t')) \in ZZ(t, t')$ . Изображение  $v$  называется допустимым, если существует допустимый вариант  $(v', s)$ . Если вариант  $(v', s)$  является допустимым, то описание  $s$  называется возможным описанием изображения  $v$ .

Двумерная грамматика  $G$  представляет собой совокупность перечисленных средств, т. е. шестерку

$$(1) \quad \langle V, S, Tv, Ts, R, \{ZZ(t, t'); (t, t') \in N\} \rangle.$$

Рассмотрим несколько примеров двумерных грамматик.

*Пример 2.* Рассмотрим двумерную грамматику, порождающую множество отягощенных изображений, рассмотренных в примере 1.

Поле описания  $Ts$  представляет собой множество пар  $t, j$  ( $j=1, 2, \dots, m$ ;  $j=1, 2, \dots, n$ ).

Поле зрения  $Tv$  — множество, изоморфное полю описания.  
 Предикат соседства  $R(t, t')$  равен единице для любой пары клеток поля описания вида  $\{(i, j), (i+1, j)\}$  или  $\{(i, j), (i, j+1)\}$ . Этот предикат

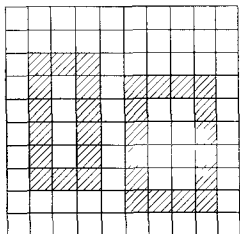


Рис. 1

имеет следующее содержание: наличие горизонтальной линии и отсутствие вертикальной.

Множество  $ZZ(t, t')$  для соседних клеток  $t \in Ts$  и  $t' \in Tv$  имеет вид

$$ZZ(t, t') = \{(B\Gamma, 1), (\overline{B\Gamma}, 1), (B\overline{\Gamma}, 1), (\overline{B\overline{\Gamma}}, 0)\}.$$

Это свидетельствует о том, что в клетке поля зрения записан 0 (т. е. она является белой), лишь в том случае, если через нее не проходит ни горизонтальная, ни вертикальная линия.

Для пары  $\{(i, j), (i, j+1)\}$  соседних клеток поля описания множество  $ZZ(t, t')$  имеет вид

$$ZZ(t, t') = \{(B\Gamma, B\overline{\Gamma}), (\overline{B\Gamma}, B\overline{\Gamma}), (B\overline{\Gamma}, \overline{B\overline{\Gamma}}), (\overline{B\overline{\Gamma}}, \overline{B\overline{\Gamma}}), (B\overline{\Gamma}, B\Gamma), (B\Gamma, \overline{B\overline{\Gamma}}), (\overline{B\overline{\Gamma}}, B\Gamma), (B\overline{\Gamma}, \overline{B\Gamma})\},$$

что приводит к невозможности такой ситуации, когда через какую-то клетку определенной строки проходит горизонтальная линия, а через другую клетку этой же строки горизонтальная линия не проходит.

Для того чтобы исключить аналогичную ситуацию для вертикальных линий, множество  $ZZ(t, t')$  для пары клеток вида  $\{(i, j), (i+1, j)\}$  должно иметь вид

$$ZZ\begin{pmatrix} t \\ t' \end{pmatrix} = \left\{ \begin{pmatrix} B\Gamma \\ B\Gamma \end{pmatrix}, \begin{pmatrix} B\Gamma \\ \overline{B\Gamma} \end{pmatrix}, \begin{pmatrix} \overline{B\Gamma} \\ \overline{B\Gamma} \end{pmatrix}, \begin{pmatrix} \overline{B\Gamma} \\ B\Gamma \end{pmatrix}, \begin{pmatrix} B\overline{\Gamma} \\ B\overline{\Gamma} \end{pmatrix}, \begin{pmatrix} B\overline{\Gamma} \\ \overline{B\overline{\Gamma}} \end{pmatrix}, \begin{pmatrix} \overline{B\overline{\Gamma}} \\ \overline{B\overline{\Gamma}} \end{pmatrix}, \begin{pmatrix} \overline{B\overline{\Gamma}} \\ B\overline{\Gamma} \end{pmatrix} \right\}$$

Для большей наглядности в последней записи символы расположены один над другим, так как клетки  $t$  и  $t'$  являются соседними, причем такими, что клетка  $t$  находится над клеткой  $t'$ .

*Пример 3.* В этом примере будет рассмотрена грамматика, порождающая либо изображение  $v(i, j) = 0$ , либо изображения, которые представляют собой совокупность прямоугольников произвольных размеров, составленных из горизонтальных и вертикальных линий. Эти прямоугольники произвольно расположены в поле зрения, но не соприкасаются один с другим, и при этом ни один прямоугольник не находится внутри другого. Пример такого изображения дан на рис. 1.

Поле зрения, поле описания, предикат соседства и алфавит сигналов выбраны так же, как и в примере 2.

Алфавит структурных элементов состоит из символов Л (левая сторона), П (правая сторона), Н (нижняя сторона), В (верхняя сторона), ЛН (левый нижний угол), ПН (правый нижний угол), ПВ (правый

верхний угол), ВНЕШ (точки, не лежащие внутри прямоугольника), ВНУТ (точки, лежащие внутри прямоугольника), ЛВ (левый верхний угол).

Множество  $ZZ(t, t')$  имеет вид:  
для клеток  $t \in Ts, t' \in Tv, (t, t') \in N$

$$ZZ(t, t') = \{(\text{Л}, 1), (\text{П}, 1), (\text{В}, 1), (\text{Н}, 1), (\text{ЛН}, 1), (\text{ЛВ}, 1), (\text{ПН}, 1), (\text{ПВ}, 1), (\text{ВНЕШ}, 0), (\text{ВНУТ}, 0)\};$$

для клеток  $t = (i, j), t' = (i, j+1)$ , т. е. клеток поля описания, соседних по горизонтали, причем клетка  $t$  находится слева от клетки  $t'$ ,

$$ZZ(t, t') = \{(\text{ВНЕШ}, \text{ВНЕШ}), (\text{ВНЕШ}, \text{ЛВ}), (\text{ВНЕШ}, \text{ЛН}), (\text{ВНЕШ}, \text{ЛВ}), (\text{ЛВ}, \text{ВНУТ}), (\text{ЛН}, \text{Н}), (\text{В}, \text{В}), (\text{ВНУТ}, \text{ВНУТ}), (\text{Н}, \text{Н}), (\text{В}, \text{ПВ}), (\text{ВНУТ}, \text{П}), (\text{Н}, \text{ПН}), (\text{ПВ}, \text{ВНЕШ}), (\text{П}, \text{ВНЕШ}), (\text{ПН}, \text{ВНЕШ})\};$$

для клеток  $t = (i, j), t' = (t+1, j)$ , т. е. для соседних по вертикали клеток, причем клетка  $t$  находится над клеткой  $t'$ ,

$$ZZ \begin{pmatrix} t \\ t' \end{pmatrix} = \left\{ \begin{pmatrix} \text{ВНЕШ} \\ \text{ВНЕШ} \end{pmatrix}, \begin{pmatrix} \text{ВНЕШ} \\ \text{ЛВ} \end{pmatrix}, \begin{pmatrix} \text{ВНЕШ} \\ \text{В} \end{pmatrix}, \begin{pmatrix} \text{ВНЕШ} \\ \text{ПВ} \end{pmatrix}, \begin{pmatrix} \text{ЛВ} \\ \text{Л} \end{pmatrix}, \begin{pmatrix} \text{В} \\ \text{ВНУТ} \end{pmatrix}, \begin{pmatrix} \text{ПВ} \\ \text{П} \end{pmatrix}, \begin{pmatrix} \text{Л} \\ \text{Л} \end{pmatrix}, \begin{pmatrix} \text{ВНУТ} \\ \text{ВНУТ} \end{pmatrix}, \begin{pmatrix} \text{П} \\ \text{П} \end{pmatrix}, \begin{pmatrix} \text{Л} \\ \text{ЛН} \end{pmatrix}, \begin{pmatrix} \text{ВНУТ} \\ \text{Н} \end{pmatrix}, \begin{pmatrix} \text{П} \\ \text{ПН} \end{pmatrix}, \begin{pmatrix} \text{ЛН} \\ \text{ВНЕШ} \end{pmatrix}, \begin{pmatrix} \text{Н} \\ \text{ВНЕШ} \end{pmatrix}, \begin{pmatrix} \text{ПН} \\ \text{ВНЕШ} \end{pmatrix} \right\}.$$

Описание примера 3 закончено. В работе [7] рассмотрен более полный список примеров, иллюстрирующих богатые возможности двумерных грамматик.

Перейдем к изложению метода двумерного программирования — метода анализа изображений, порождаемых двумерными грамматиками.

Пусть  $X$  — множество реальных сигналов,  $x: Tv \rightarrow X$  — реальное изображение;  $F(v, x)$  — функция сходства, которая имеет вид

$$(2) \quad F(v, x) = \sum_{t \in T_1} f(v(t), x).$$

Задачей распознавания и описания в двумерной грамматике  $G$  изображения  $x$  будем называть задачу отыскания такого изображения  $v^*$ , порождаемого двумерной грамматикой  $G$ , которое обеспечивает максимум функции сходства, и отыскания возможного описания  $s^*$  изображения  $v^*$ .

Для изложения метода решения этой задачи необходимо ввести дополнительные понятия.

Через  $Z(t)$ ,  $t \in T$  обозначим множество символов  $z$ , допустимых в клетке  $t$ .

Пару  $(z, t)$ , такую, что  $z \in Z(t)$ , будем обозначать  $\sigma$  и называть вершиной. Зададим функции  $z(\sigma) = z(z, t) = z$  и  $t(\sigma) = t(z, t) = t$ . Будем говорить, что  $z(\sigma)$  является наименованием вершины  $\sigma$ , а  $t(\sigma)$  — клеткой, в которой она лежит. Множество вершин, лежащих в клетке  $t$ , обозначим через  $\bar{\sigma}(t)$ . Через  $\bar{\sigma}\bar{\sigma}(t', t'')$  обозначим множество пар  $(\sigma_1, \sigma_2)$ , таких, что  $t(\sigma_1) = t', t(\sigma_2) = t''$ , а  $(z(\sigma_1), z(\sigma_2)) \in ZZ(t', t'')$ . Элемент  $(\sigma, \sigma')$  множества  $\bar{\sigma}\bar{\sigma}(t, t')$  будем называть дужкой, соединяющей вершины  $\sigma$  и  $\sigma'$ .

Пусть  $M(t)$  — множество клеток, соседних с клеткой  $t$ . Разделим множество  $M(t)$  на два непересекающихся подмножества  $M^+(t)$  и  $M^-(t)$ , удовлетворяющих следующему условию: если  $\tau \in M^+(t)$ , то  $t \in M^-(\tau)$ , и наоборот, если  $\tau \in M^-(t)$ , то  $t \in M^+(\tau)$ . Каждой паре  $(\sigma, \tau)$ ,  $(\sigma \in \bar{\sigma}(t))$ ,

$t \in T$ ,  $\tau \in M(t(\sigma))$ ) поставим в соответствие вещественное число  $\varphi_\sigma^t$ , которое назовем потенциалом. Каждой вершине  $\sigma$  поставим в соответствие число

$$(3) \quad \Delta(\sigma) = \sum_{\tau \in M^+(t(\sigma))} \varphi_\sigma^{\tau} - \sum_{\tau \in M^-(t(\sigma))} \varphi_\sigma^{\tau},$$

которое назовем напряжением на вершине  $\sigma$ . Каждой паре вершин  $(\sigma, \sigma')$ , соединенных дужкой, т. е. таких, что  $(t(\sigma), t(\sigma')) \in N$ ,  $(\sigma, \sigma') \in \equiv \sigma \sigma'(t(\sigma), t(\sigma'))$ , поставим в соответствие число  $\delta(\sigma, \sigma')$ , которое назовем напряжением между вершинами  $\sigma$  и  $\sigma'$  либо напряжением на дужке  $(\sigma, \sigma')$ . Напряжение  $\delta(\sigma, \sigma')$  определяется выражением

$$(4) \quad \delta(\sigma, \sigma') = \varphi_\sigma^{t(\sigma')} - \varphi_{\sigma'}^{t(\sigma)}, \text{ если } t(\sigma) \in M^+(t(\sigma'));$$

$$(5) \quad \delta(\sigma, \sigma') = \varphi_{\sigma'}^{t(\sigma)} - \varphi_\sigma^{t(\sigma')}, \text{ если } t(\sigma') \in M^+(t(\sigma)).$$

По определению ясно, что  $\delta(\sigma, \sigma') = -\delta(\sigma', \sigma)$ .

Высотой вершины  $\sigma$  назовем величину  $h(\sigma)$ , определяемую формулой

$$(6) \quad h(\sigma) = f(z(\sigma), x) - \Delta(\sigma), \text{ если } t(\sigma) \in Tv;$$

$$(7) \quad h(\sigma) = -\Delta(\sigma), \text{ если } t(\sigma) \in Ts.$$

Потенциальной функцией назовем функцию

$$(8) \quad K(\varphi, x) = \sum_{\sigma \in T} \max_{\sigma \in \sigma(t)} h(\sigma).$$

Допустимым набором  $\Phi$  потенциалов  $\varphi_\sigma^t$  будем называть набор, удовлетворяющий условиям  $\delta(\sigma, \sigma') \geq 0$  для всех пар  $(\sigma, \sigma')$ , таких, что  $(z(\sigma), z(\sigma')) \in ZZ(t(\sigma), t(\sigma'))$ ,  $R(t(\sigma), t(\sigma')) = 1$ .

Пусть дана грамматика  $G$  и набор  $\Phi$  потенциалов, допустимый в этой грамматике. Через  $G'(\Phi)$  будем обозначать грамматику, являющуюся подграмматикой в  $G$  и получающуюся из нее вычеркиванием из множеств  $Z(t)$ ,  $t \in T$  тех символов  $z$ , для которых

$$(9) \quad h(z, t) \neq \max_{z \in z(t)} h(z, t),$$

а также вычеркиванием из множества  $ZZ(t, t')$  тех пар  $(z, z')$ , для которых

$$(10) \quad \delta((z, t), (z', t')) \neq 0.$$

Грамматика является противоречивой, если множество порождаемых ею допустимых изображений является пустым.

В работе [7] доказаны теоремы.

*Теорема 1.* Если  $\Phi$  является допустимым набором потенциалов в грамматике  $G$ , то любое изображение, порождаемое грамматикой  $G'(\Phi)$ , обеспечивает максимум функции сходства на множестве изображений, порождаемых грамматикой  $G$ .

*Теорема 2.* Если существует набор  $\Phi$ , такой, что грамматика  $G'(\Phi)$  непротиворечива, то этот набор обеспечивает минимум потенциальной функции.

Эти две теоремы указывают путь решения задачи распознавания реальных сигналов. Он заключается в отыскании потенциалов  $\Phi$ , минимизирующих потенциальную функцию, и последующем нахождении допустимого изображения в грамматике  $G'(\Phi)$ .

Ниже описан алгоритм, реализующий (с определенными недостатками, которые будут прокомментированы) указанный путь. Общая схема этого алгоритма заключается в следующем.

Алгоритм выполняет последовательность операций, на каждой из которых происходит переход от одного допустимого значения набора  $\Phi$  потенциалов к другому допустимому значению. Пусть  $\Phi^{(i)}$  — некоторое очередное значение набора  $\Phi$ . Для этого значения строится грамматика  $G''(\Phi^{(i)})$ , являющаяся подграмматикой исходной грамматики  $G$  и несущественно (на данном этапе изложения) отличающаяся от введенной выше грамматики  $G'(\Phi^{(i)})$ . Для грамматики  $G''(\Phi^{(i)})$  производится тест на существование в этой грамматике допустимого изображения. Это тест заключается в последовательном эквивалентном преобразовании грамматики  $G''(\Phi^{(i)})$ . Преобразования являются эквивалентными в том смысле, что они не изменяют множества порождаемых изображений. Последовательность этих преобразований заканчивается за конечное число шагов получением грамматики  $G''(\Phi^{(i)})$ , для которой либо выполняются некоторые вполне очевидные условия ее противоречивости, либо не выполняются.

В последнем случае происходит останов алгоритма.

Если же доказана противоречивость грамматики  $G''(\Phi^{(i)})$ , а следовательно, и грамматики  $G'(\Phi^{(i)})$ , то тем самым доказано, что набор  $\Phi^{(i)}$  не удовлетворяет условиям теоремы 2. В этом случае последовательность эквивалентных преобразований грамматики, приведенная от грамматики  $G''(\Phi^{(i)})$  к грамматике  $G''(\Phi^{(i)})$ , используется для построения вектора  $\Delta\Phi^{(i+1)}$ . Свойства этого вектора таковы, что по крайней мере для малых положительных  $\alpha$  набор  $\Phi^{(i)} + \Delta\Phi^{(i+1)}\alpha$  является допустимым, а значение потенциальной функции  $K(\Phi^{(i)} + \Delta\Phi^{(i+1)}\alpha)$  строго меньше значения  $K(\Phi^{(i)})$ . Значение  $\Phi^{(i)} + \Delta\Phi^{(i+1)}\alpha$  объявляется следующим очередным значением  $\Phi^{(i+1)}$  набора потенциалов  $\Phi$ .

Приступим к подробному изложению алгоритма.

### 3. Описание алгоритма

Множество  $\bigcup_{t \in T} \bar{\sigma}(t)$  обозначим через  $\bar{\sigma}$ . Вычеркиванием вершины называется процесс изменения некоторой функции  $c(\sigma)$ , заданной на множестве вершин. Эта функция для любой вершины  $\sigma \in \bar{\sigma}(t)$ ,  $t \in T$  принимает значения из множества  $\{0, \text{нл}\} \cup M(t)$ . Первоначально функция  $c(\sigma)$  для всех  $\sigma \in \bar{\sigma}$  равна нулю, и это значит, что все вершины не вычеркнуты. Если  $c(\sigma) \neq 0$ , то это значит, что вершина  $\sigma$  вычеркнута, причем значение функции  $c(\sigma)$  в этом случае называется причиной вычеркивания.

Для каждого реального сигнала  $x$  и грамматики  $G$  работа алгоритма состоит в однократном выполнении предварительной итерации и многократном, вообще говоря, выполнении общей итерации.

#### 3.1. Предварительная итерация

Всем потенциалам  $\Phi$  и направлениям  $\Delta\Phi$  присваиваются нулевые значения. Все вершины полагают невычеркнутыми, т. е.  $c(\sigma) = 0$  для всех  $\sigma \in \bar{\sigma}$ . Определяются значения высот вершин:

$$(11) \quad \begin{aligned} h(\sigma) &= f(z(\sigma), x), & \text{если } t(\sigma) \in Tv; \\ h(\sigma) &= 0, & \text{если } t(\sigma) \in Ts, \text{ поскольку } \Delta(\sigma) = 0. \end{aligned}$$

#### 3.2. Общая итерация

3.2.1. Поиск направления допустимого убывания потенциальной функции  $K(\Phi, x)$ , т. е. нахождения величины  $\Delta\Phi$

Построение грамматики  $G''(\Phi)$ . Для любой вершины  $\sigma \in \bar{\sigma}$ , для которой выполняется соотношение

$$(12) \quad h(\sigma^*) < \max_{\sigma \in \bar{\sigma}(t(\sigma^*))} h(\sigma) - p_1$$

производится вычеркивание, называемое вычеркиванием по нулевой причине, т. е. метке  $c(\sigma')$  присваивается значение нл.

Следует обратить внимание на различие между условием вычеркивания (12) и условием (9), которое было введено для формулировки теоремы 1. При формировании грамматики  $G'(\Phi)$ , обладающей свойствами, указанными в теореме 1, вычеркиванию по нулевой причине подлежат все вершины, высоты которых не являются максимальными в данной клетке. В процессе же работы алгоритма применяется особое вычеркивание, называемое  $p$ -вычеркиванием. При этом вычеркиванию по нулевой причине подлежат не все не максимальные вершины, а лишь те, высоты которых отличаются более чем на величину  $p_1$  от максимальной высоты в данной клетке.

Аналогичным образом условие вычеркивания дужки в процессе работы алгоритма отличается от введенного ранее условия вычеркивания (10), формирующего грамматику  $G'(\Phi)$ . Вычеркиванию подлежат не все дужки, напряжение на которых не равно нулю, а лишь те из них, напряжение на которых удовлетворяет условию:

$$(13) \quad \delta(\sigma, \sigma') > p_2.$$

где  $(\sigma, \sigma') \in \overline{\sigma\sigma}$ , а  $\overline{\sigma\sigma} = \bigcup_{(t,t') \in \gamma} \overline{\sigma\sigma}(t, t')$  — множество всех дужек.

Величины  $p_1$  и  $p_2$  являются важными параметрами алгоритма, которые, как будет показано далее, определяют скорость его сходимости и точность полученного решения. Величина  $p_1$  называется порогом вычеркивания вершин, а величина  $p_2$  — порогом вычеркивания дужек.

*Эквивалентные преобразования грамматики  $G''(\Phi)$ .* Пусть  $\Omega'(\sigma, \tau)$  — множество вершин, находящихся в клетке  $\tau$ , и каждая вершина из этого множества соединена невычеркнутой дужкой с вершиной  $\sigma$ , т. е.  $\sigma' \in \overline{\sigma\sigma}(\sigma, \tau)$ , если  $(\sigma, \sigma') \in \overline{\sigma\sigma}(t(\sigma), \tau)$ ,  $\delta(\sigma, \sigma') \leq p_2$  и  $\sigma' \in \sigma(\tau)$ .

Для любой клетки  $t \in T$  вершина  $\sigma \in \overline{\sigma\sigma}(t)$  называется вычеркнутой по причине  $t \in M(t)$ , если все вершины в множестве  $\Omega'(\sigma, \tau)$  оказались вычеркнутыми по какой-либо причине либо если  $\Omega'(\sigma, \tau) = \emptyset$ .

Вычеркивание вершины  $\sigma$  по причине  $t$  выражается в том, что метке  $c(\sigma)$  присваивается значение  $t$ . Операция вычеркивания по нулевой причине и  $t$  применяется только к невычеркнутым вершинам, которые в результате этой операции становятся вычеркнутыми.

Операция вычеркивания вершин множества  $\bar{\sigma}$  применяется до тех пор, пока не выполнятся одно из следующих условий:

а) при вычеркивании некоторой вершины  $\sigma$  все вершины множества  $\bar{\sigma}(t(\sigma))$  оказались вычеркнутыми; клетку  $t(\sigma)$ , в которой вычеркнуты все вершины множества  $\bar{\sigma}(t(\sigma))$ , обозначим через  $t^0$  и назовем нуль-клеткой;

б) ни одна вершина множества  $\sigma$  не может быть вычеркнута по причине  $t$ , при этом в каждом множестве вершин  $\bar{\sigma}(t)$ ,  $t \in T$  имеется хотя бы одна невычеркнутая вершина.

Если выполняется условие а), то приступаем к определению направления  $\Delta\Phi$ . Выполнение этого условия доказывает противоречивость грамматики  $G''(\Phi)$ .

В случае выполнения условия б) производится остановка алгоритма. *Определение направления  $\Delta\Phi$ .* Пусть в результате вычеркивания вершин все вершины клетки  $t^0$  оказались вычеркнутыми. В таком случае в нуль-клетке обязательно найдется по крайней мере одна вершина, вычеркнутая по какой-либо причине  $t \in M(t^0)$ , поскольку согласно правилам вычеркивания вершин все вершины множества  $\bar{\sigma}(t)$  любой клетки  $t \in T$  не могут быть вычеркнуты по нулевой причине. Для любой вершины  $\sigma \in \bar{\sigma}(t^0)$ , вычеркнутой по какой-либо причине  $t \in M(t^0)$ , производится присваивание величине  $\Delta\Phi$  следующих значений:

$$\Delta\varphi_\sigma = +1, \quad \text{если } \tau \in M^+(t^0);$$

$$\Delta\varphi_\sigma = -1, \quad \text{если } \tau \in M^-(t^0).$$

Для каждой вершины  $\sigma'$  множества  $\Omega'(\sigma, \tau)$ , определяемого для вычеркнутой по причине  $\tau$  вершины  $\sigma$  клетки  $t^0$ , определяется величина  $\Delta\varphi_{\sigma'} = \Delta\varphi_\sigma$ , где  $\tau = t(\sigma')$ ,  $t^0 = t(\sigma)$ . Каждой вершине  $\sigma' \in \Omega'(\sigma, \tau)$  клетки  $\tau \in M(t^0)$  приписывается метка  $*$ .

Вершина, имеющая метку  $*$ , называется помеченной, а вершина, не имеющая метку  $*$ , — непомеченной.

Получение из помеченной вершины непомеченной называется стиранием метки  $*$ .

Для любой помеченной вершины  $\sigma$  выполняется следующий элементарный такт.

У вершины  $\sigma$  метка стирается. Если вершина  $\sigma$  была вычеркнута по нулевой причине, то переходят к поиску другой помеченной вершины. В противном случае одновременно со стиранием метки  $*$  величине  $\Delta\varphi_\sigma$ , где  $\tau = c(\sigma)$ , присваивается такое значение, которое удовлетворяет одному из следующих двух условий:

$$(14) \quad \Delta\Delta(\sigma) = 0, \quad \text{если } t(\sigma) \neq t^0;$$

$$(15) \quad \Delta\Delta(\sigma) = +1, \quad \text{если } t(\sigma) = t^0,$$

где

$$(16) \quad \Delta\Delta(\sigma) = \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_\tau - \sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_\tau.$$

Для вершины  $\sigma$  находится множество вершин  $\Omega'(\sigma, \tau)$ , соединенных с вершиной  $\sigma$  невычеркнутыми дужками и расположенных в клетке  $\tau = c(\sigma)$ .

Для каждой вершины  $\sigma' \in \Omega'(\sigma, \tau)$  меняют величину  $\Delta\varphi_{\sigma'}$  согласно уравнению

$$(17) \quad \Delta\varphi_{\sigma'}^{i(\sigma)} = \Delta\varphi_{\sigma'}^{i(\sigma')},$$

если  $|\Delta\varphi_{\sigma'}^{i(\sigma)}| < |\Delta\varphi_{\sigma'}^{i(\sigma')}|$ , в противном случае величина  $\Delta\varphi_{\sigma'}^{i(\sigma)}$  остается прежней.

Кроме того, каждой вершине  $\sigma' \in \Omega'(\sigma, \tau)$ , для которой изменили величину  $\Delta\varphi_{\sigma'}^{i(\sigma)}$ , приписывается метка  $*$ .

Элементарный такт повторяется до тех пор, пока существует хоть одна помеченная вершина в множестве  $\mathcal{B}$ .

### 3.2.2. Вычисление шага $\alpha$

Величина шага  $\alpha$  должна выбираться так, чтобы набор потенциалов  $\varphi + \alpha\Delta\varphi$  остался допустимым, а величина потенциальной функции  $K(\varphi + \alpha\Delta\varphi, x)$  стала как можно меньше. Допустимость набора  $\varphi + \alpha\Delta\varphi$  обозначает неотрицательность напряжений на всех дужках или, иными словами, выполнение неравенства

$$(18) \quad (\varphi_\sigma^{i(\sigma')} + \alpha\Delta\varphi_\sigma^{i(\sigma')}) - (\varphi_{\sigma'}^{i(\sigma)} + \alpha\Delta\varphi_{\sigma'}^{i(\sigma)}) \geq 0$$

для всех дужек  $(\sigma, \sigma')$ , таких, что  $t(\sigma) \in M^+(t(\sigma'))$ .

Исходя из неравенства (18) следует, что величина  $\alpha$  должна удовлетворять неравенствам

$$(19) \quad \alpha \leq \frac{\varphi_\sigma^{i(\sigma')} - \varphi_{\sigma'}^{i(\sigma)}}{\Delta\varphi_{\sigma'}^{i(\sigma)} - \Delta\varphi_\sigma^{i(\sigma')}}.$$

для всех дужек  $(\sigma, \sigma')$ , таких, что  $t(\sigma) \in M^+(t(\sigma'))$ , причем вычеркнутых, т. е. удовлетворяющих условию  $\Delta\varphi_{\sigma'}^{t(\sigma)} - \Delta\varphi_{\sigma}^{t(\sigma')} > 0$ .

Условие минимальности функции  $K(\varphi + \alpha\Delta\varphi, x) = \sum_{i \in \Gamma} h_i$  по величине  $\alpha$

равносильно требованию того, чтобы при переходе от набора  $\varphi$  к набору  $\varphi + \alpha\Delta\varphi$  высота  $h_i$  клетки  $t \neq t^0$  не увеличилась, а высота  $h_{t^0}$  клетки  $t^0$  уменьшилась как можно больше. Первое из этих условий записывается в виде равенств

$$(20) \quad h_{t(\sigma)} \geq h(\sigma) - \left( \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_{\sigma'}^{\tau} - \sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} \right) \alpha; \quad \sigma \in \bar{\sigma}, \quad t(\sigma) \neq t^0,$$

эквивалентных неравенствам

$$(21) \quad \alpha \leq \frac{h_{t(\sigma)} - h(\sigma)}{\sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} - \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_{\sigma'}^{\tau}},$$

которые должны выполняться для всех вершин  $\sigma \in \bar{\sigma} (t^0)$ , таких, что выполняется неравенство  $\sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} - \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_{\sigma'}^{\tau} > 0$ , свидетельствующее

о том, что вершина  $\sigma$  вычеркнута по нулевой причине.

Для того чтобы найти величину  $\alpha$ , которая обеспечивает минимум высоты  $h_{t^0}$  клетки  $t^0$ , необходимо рассуждать следующим образом.

Выделим в клетке  $t^0$  совокупность  $\bar{\sigma}^*$  вершин, имеющих максимальную высоту при  $\alpha=0$ . Высоты всех этих вершин с ростом  $\alpha$  будут уменьшаться, но по-разному, в зависимости от величин потенциалов на этих вершинах. Высота клетки  $t^0$  при малых значениях  $\alpha$  зависит от  $\alpha$  следующим образом:

$$h_{t^0}(\alpha) = \max_{\sigma \in \bar{\sigma}^*} \left[ h(\sigma) - \alpha \left( \sum_{\tau \in M^+(t^0)} \Delta\varphi_{\sigma'}^{\tau} - \sum_{\tau \in M^-(t^0)} \Delta\varphi_{\sigma}^{\tau} \right) \right].$$

Это выражение, очевидно, эквивалентно следующему:

$$(22) \quad h_{t^0}(\alpha) = h_{t^0} - \alpha \max_{\sigma \in \bar{\sigma}^*} \left( \sum_{\tau \in M^+(t^0)} \Delta\varphi_{\sigma'}^{\tau} - \sum_{\tau \in M^-(t^0)} \Delta\varphi_{\sigma}^{\tau} \right).$$

Остальные вершины в клетке  $t^0$ , не принадлежащие  $\bar{\sigma}^*$ , т. е. вычеркнутые по нулевой причине, либо увеличивают свою высоту с ростом  $\alpha$ , либо не изменяют ее

$$(23) \quad h_{\alpha}(\sigma, \alpha) = h(\sigma) - \alpha \left( \sum_{\tau \in M^+(t^0)} \Delta\varphi_{\sigma'}^{\tau} - \sum_{\tau \in M^-(t^0)} \Delta\varphi_{\sigma}^{\tau} \right)$$

для всех  $\sigma \in \bar{\sigma}(t)$ ,  $\sigma \notin \bar{\sigma}^*$ .

Величина  $\alpha$ , очевидно, должна удовлетворять условию  $h_{t^0}(\alpha) \geq h_{\alpha}(\sigma, \alpha)$ , которое должно выполняться для всех  $\sigma \in \bar{\sigma}(t^0)$ ,  $\sigma \notin \bar{\sigma}^*$ , или, в иной записи,

$$(24) \quad \alpha \leq \frac{h_{t^0} - h(\sigma)}{\max_{\sigma \in \bar{\sigma}^*} \left( \sum_{\tau \in M^+(t^0)} \Delta\varphi_{\sigma'}^{\tau} - \sum_{\tau \in M^-(t^0)} \Delta\varphi_{\sigma}^{\tau} \right) - \left( \sum_{\tau \in M^+(t^0)} \Delta\varphi_{\sigma'}^{\tau} - \sum_{\tau \in M^-(t^0)} \Delta\varphi_{\sigma}^{\tau} \right)},$$

$\bar{\sigma} \in \bar{\sigma}(t^0)$ ,  $\bar{\sigma} \notin \bar{\sigma}^*$ .

Значение шага  $\alpha$  должно быть равно максимально возможному из удовлетворяющих неравенствам (24), (21), (19).

### 3. 2. 3. Нахождение измененных значений потенциалов $\Phi$

Обозначим через  $\alpha^{(i)}$ ,  $\varphi_\sigma^{(i)}$ ,  $h^{(i)}(\sigma)$ ,  $\Delta\Delta^{(i)}(\sigma)$ ,  $\Delta\varphi_\sigma^{(i)}$  соответственно значения величин  $\alpha$ ,  $\varphi_\sigma$ ,  $h(\sigma)$ ,  $\Delta\Delta(\sigma)$ ,  $\Delta\varphi_\sigma$ , найденных на  $i$ -й итерации.

Для любой вершины  $\sigma \in \bar{\sigma}$  и  $\tau \in M(i(\sigma))$ , для которых на  $i$ -й итерации величина  $\Delta\varphi_\sigma^{(i)} \neq 0$ , определяется значение потенциала

$$(25) \quad \varphi_\sigma^{(i)} = \varphi_\sigma^{(i-1)} + \alpha^{(i)} \Delta\varphi_\sigma^{(i)}.$$

Для любой вершины  $\sigma \in \bar{\sigma}$ , для которой приращение напряжения  $\Delta\Delta^{(i)}(\sigma) \neq 0$ , определяется значение высоты

$$(26) \quad h^{(i)}(\sigma) = h^{(i-1)}(\sigma) - \alpha^{(i)} \Delta\Delta^{(i)}(\sigma).$$

Значение функции  $c(\sigma)$  для каждой вершины  $\sigma \in \bar{\sigma}$  полагают равным нулю, т. е. вершины делаются невычеркнутыми. Значения величин  $\Delta\varphi_\sigma^{(i)}$  для всех вершин  $\sigma \in \bar{\sigma}$  и величин  $\tau \in M(i(\sigma))$  приравняются нулю.

Пример работы алгоритма приведен в приложении.

## 4. Исследование алгоритма

Результатом теоретического исследования алгоритма являются формулируемые ниже теоремы. Теоремы 3, 4 и леммы 1, 2 констатируют динамические свойства, т. е. его сходимость. Теорема 5 определяет свойства того состояния алгоритма, в котором происходит его останов.

Леммы 1, 2 и доказательства всех утверждений приводятся в приложении.

*Теорема 3.* На каждой итерации алгоритма поиск возможного направления  $\Delta\Phi$  производится за конечное число шагов.

Кроме поиска направления  $\Delta\Phi$ , в одну итерацию алгоритма входит построение и преобразование грамматики  $G''(\Phi)$ , предшествующей поиску направления, и вычисление величины шага и очередного значения набора потенциалов. Является очевидным, что для реализации этих вычислений также требуется конечное число итераций.

*Теорема 4.* Пусть функция  $f(v, x)$  ограничена сверху и снизу, а грамматика  $G$  непротиворечива. В таком случае остановка алгоритма произойдет после конечного числа итераций.

Пусть  $\tilde{\Phi}$  — набор потенциалов, полученный после окончания работы алгоритма, а  $G''(\tilde{\Phi})$  — соответствующая грамматика. Пусть  $\tilde{v}$  — любое изображение в этой грамматике. Если бы грамматика  $G''(\tilde{\Phi})$  совпала с грамматикой  $G'(\tilde{\Phi})$ , то изображение  $\tilde{v}$  обеспечивало бы максимум функции сходимости на множестве изображений, порождаемых грамматикой  $G$ . Однако грамматика  $G'(\tilde{\Phi})$  является подграмматикой грамматики  $G''(\tilde{\Phi})$ , поэтому последняя может, вообще говоря, включать в себя и изображения, не являющиеся оптимальными по значению функции сходимости. Свойства изображений, являющихся допустимыми в  $G''(\tilde{\Phi})$ , выражаются следующей теоремой.

*Теорема 5.* Пусть  $\tilde{\Phi}$  — допустимый в грамматике  $G$  набор потенциалов;  $\tilde{v}$  — допустимое изображение в грамматике  $G''(\tilde{\Phi})$ ;  $v^*$  — допустимое изображение в грамматике  $G$ , максимизирующее функцию сходимости,  $\Psi = \max_{t \in T} |M(t)|$ . В таком случае

$$F(v^*, x) - F(\tilde{v}, x) \leq p_1 |T| + p_2 |T| \Psi / 2$$

### 5. Экспериментальные исследования алгоритма

Выполненное теоретическое исследование оставляет открытыми два вопроса.

1. Предлагаемый в работе алгоритм обладает тем свойством, что он в некоторых ситуациях указывает изображение, и в таком случае это изображение является допустимым в данной грамматике, а функции схождения для этого изображения отличается от максимально возможной на некоторую величину, которую надлежащим выбором величин  $p_1$  и  $p_2$  можно сделать сколь угодно малой. Эта ситуация имеет место, когда в грамматике  $G''(\Phi)$ , где  $\Phi$  — набор потенциалов, полученный на последней итерации алгоритма, содержится допустимое изображение. Если же грамматика  $G''(\Phi)$  такова, что в ней не содержится ни одного допустимого изображения, то алгоритм выдает ответ, который следует понимать, как отказ от распознавания. В любом случае алгоритмом гарантируется отсутствие ошибочных решений, т. е. указание изображений, функция схождения которых отличается от максимально возможной более, чем на заданную величину. В то же время не ясно, насколько часто алгоритм принимает решение «отказ от распознавания».

2. Хотя в работе и доказано, что алгоритм сходится за конечное число шагов, однако не ясно, чему равно это количество шагов в реальных ситуациях.

Для выяснения этих вопросов была предпринята экспериментальная проверка алгоритма.

При проведении экспериментальных исследований алгоритма использовались двумерные грамматики, рассмотренные в примерах 2 и 3, а также двумерные грамматики, приведенные ниже.

*Пример 4.* Рассмотрим двумерную грамматику, порождающую множество эталонных изображений, любое из которых определяет на поле зрения два подмножества: подмножество белых и подмножество черных клеток. Черная клетка, имеющая по соседству белую клетку, называется контурной. Любая строка клеток должна содержать не более одной контурной клетки.

Для меньшей громоздкости дальнейшего изложения введем еще одно ограничение, а именно: слева от контурной клетки должна находиться белая клетка.

Поле зрения, поле описания, предикат соседства и алфавит сигналов выбраны так же, как и в примере 2.

Алфавит структурных элементов состоит из символов Б (белая клетка), К (черная клетка, через которую проходит контур), Ч (черная клетка, через которую не проходит контур).

Множество  $ZZ(t, t')$  имеет вид:  
для клеток  $t \in Ts, t' \in Tv, (t, t') \in N$

$$ZZ(t, t') = \{(B, 0), (K, 1), (Ч, 1)\};$$

для клеток  $t = (i, j), t' = (i, j+1)$ , т. е. клеток поля описания, соседних по горизонтали, причем клетка  $t$  находится слева от клетки  $t'$ ,

$$ZZ(t, t') = \{(Б, Б), (Б, К), (К, Ч), (Ч, Ч)\};$$

для клеток  $t = (i, j), t' = (i+1, j)$ , т. е. для соседних по вертикали клеток, причем клетка  $t$  находится над клеткой  $t'$ ,

$$ZZ \begin{pmatrix} t \\ t' \end{pmatrix} = \left\{ \begin{pmatrix} Б \\ Б \end{pmatrix}, \begin{pmatrix} Б \\ К \end{pmatrix}, \begin{pmatrix} К \\ Б \end{pmatrix}, \begin{pmatrix} К \\ К \end{pmatrix}, \begin{pmatrix} Ч \\ Ч \end{pmatrix}, \begin{pmatrix} Ч \\ К \end{pmatrix}, \begin{pmatrix} Ч \\ Б \end{pmatrix} \right\}.$$

Рассматриваемая грамматика является простой в том отношении, что запоминание ее требует небольшого объема памяти. В то же время эта грамматика задает весьма обширное множество, содержащее приблизительно  $m \cdot 3^n$  изображений, где  $m$  и  $n$  — горизонтальный и вертикальный размеры

поля зрения. Будем говорить, что приведенная в данном примере грамматика формирует понятие «сконтур».

*Пример 5.* Рассмотрим двумерную грамматику, порождающую либо изображение  $v(i, j) \equiv 0$ , либо стилизованные буквы П и О, произвольных размеров, и произвольно расположенные в поле зрения, не соприкасающиеся между собой, и не находящиеся одна внутри другой.

Рассматриваемая грамматика отличается от грамматики, приведенной в примере 3, следующим:

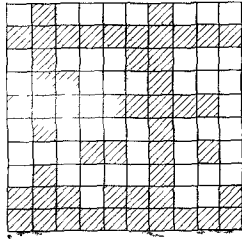


Рис. 2

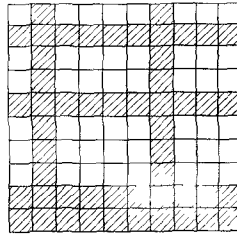


Рис. 3

алфавит структурных элементов дополнен символом М (точки, лежащие в горизонтальной строке между левым нижним и правым нижним концами буквы П);

множество  $ZZ(t, t')$  дополнено следующими парами структурных элементов:

для клеток  $t \in Ts, t' \in Tv, (t, t') \in N - \{(M, O)\}$ ;

для клеток  $t = (i, j), t' = (i, j+1) - \{(ЛН, М), (М, М), (М, ПН)\}$ ;

для клеток  $t = (i, j), t' = (i+1, j) - \left\{ \left( \begin{array}{c} \text{ВНУТ} \\ \text{М} \end{array} \right), \left( \begin{array}{c} \text{М} \\ \text{ВНЕШ} \end{array} \right) \right\}$ .

Описание примера 5 закончено.

Методика экспериментальных исследований состояла в программной реализации описанного алгоритма на ЭВМ БЭСМ-6. Распознавание изображения задавались на прямоугольном поле зрения размером не более  $m \times n = 200$ .

Опытная проверка алгоритма проводилась для некоторого изображения  $x$ , являющегося результатом случайного искажения эталонного изображения, порождаемого одной из грамматик примеров 2–5. Искажение эталонного изображения выразилось в том, что если для некоторой пары  $(i, j)$  значение сигнала  $v(i, j) = a(a=1, 0)$ , то в искаженном изображении для этой пары величина  $x(i, j)$  равна  $1-a$  с вероятностью  $p$  и  $x(i, j)$  равно  $a$  с вероятностью  $1-p$ . Величина задаваемой вероятности варьировалась от 0,05 до 0,1.

Суть опытной проверки алгоритма состояла в следующем. Для некоторого изображения  $x$  и грамматики  $G$  производился поиск набора потенциалов  $\varphi$ , такой, что грамматика  $G''(\varphi)$  содержала наименьшую часть. В грамматике  $G''(\varphi)$  затем производился поиск любого допустимого изображения и его возможного описания.

На рис. 2 приведено изображение, являющееся результатом искажения некоторого изображения, порождаемого грамматикой примера 2. На рис. 3 приведено данное изображение, а на рис. 4 приведено его описание, которое было найдено в результате работы алгоритма.

Общее количество использованных в эксперименте изображений равнялось 200. При распознавании этих изображений решение «отказ от распознавания» не было принято ни разу. Изображения, найденные в процессе работы алгоритма, обеспечивали значение функции сходства, в точности равное максимально возможному.

$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}G$	$B\bar{G}$	$\bar{B}\bar{G}$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}G$	$B\bar{G}$	$\bar{B}\bar{G}$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}G$	$B\bar{G}$	$\bar{B}\bar{G}$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}G$	$B\bar{G}$	$\bar{B}\bar{G}$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}G$	$B\bar{G}$	$\bar{B}\bar{G}$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$
$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$	$BG$	$\bar{B}\bar{G}$	$B\bar{G}$	$\bar{B}G$

Рис 4

Количество итераций алгоритма зависит от вероятности  $p$ , грамматики  $G$ , величин порогов  $p_1$  и  $p_2$ . Так, для грамматики, приведенной в примере 5, в которой размер поля  $|T|=400$ , и для значений порогов  $p_1=p_2=0,1 \max_{v(t) \in Z(t)} f(v(t), x)$ ,  $t \in Tv$  имеем, в зависимости от вероятности  $p$ , следующее среднее количество итераций  $I$ :

если  $p=0$ , то  $I=1$ ;  
 если  $p=0,05$ , то  $I=15$ ;  
 если  $p=0,1$ ; то  $I=40$ .

Эти выводы, будучи полученными экспериментально, не должны переноситься на другие неизученные ситуации.

Авторы благодарят В. А. Ковалевского и сотрудников руководимого им отдела, многочисленные обсуждения с которыми способствовали более ясному пониманию проблемы.

#### ПРИЛОЖЕНИЕ

*Пример работы алгоритма.* Рис 5 — рис 9 иллюстрируют работу описанного алгоритма. На рис. 5 изображено поле  $T$ , состоящее из 9 клеток. В каждой клетке расположены две или три вершины графа, обозначенные кружками. По рисунку ясно, какие пары клеток являются соседними, а какие нет. Рядом с клетками проставлены знаки плюс и минус, которые иллюстрируют разделение множеств  $M(t)$ ,  $t \in T$  на подмножества  $M^+(t)$  и  $M^-(t)$ . Так, например, самая крайняя левая клетка  $t \in M^+(\tau)$ , а соседняя ей клетка  $\tau \in M^-(t)$ .

Значение высоты  $h(\sigma)$  вершины  $\sigma$  записывается внутри соответствующего этой вершине круга, а значение потенциала  $\varphi_\sigma \neq 0$  для пары  $\begin{pmatrix} \tau \\ \sigma \end{pmatrix}$  записывается вне это

го круга, причем со стороны клетки  $\tau$ .

Первоначальные значения высот вершин, т. е. величины  $f(\sigma, x)$  и набор потенциалов, приведены на рис 5. Эти значения таковы, что если для каждой клетки  $t \in T$  выбрать по одной вершине  $\sigma$  максимальным значением высоты для своей клетки,

т. е. выбрать множество вершин  $B\sigma$ , то  $\sum_{\sigma \in B\sigma} h(\sigma)$  будет иметь максимальное значение, но при этом множество вершин  $B\sigma$  не образует допустимого варианта. На

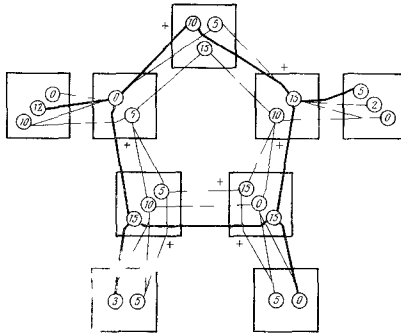


Рис 5

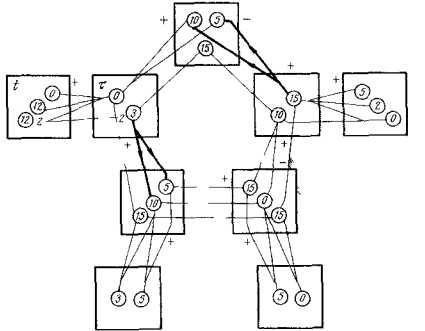


Рис 6

рис 5-8 показаны отдельные этапы процесса изменения потенциалов а следовательно и высот. На этих рисунках некоторые дужки выделены толстыми линиями. Стрелки на выделенных дужках обозначают следующее: потенциал находящийся в начале дужки изменяется так, чтобы уменьшить высоту «своей» вершины, изменение же потенциала на конце дужки увеличивает высоту соответствующей вершины. Таким образом, уменьшается высота вершины, из которой дужка только выходит, увеличивается высота вершины в которую дужка только входит и остаются неизменными высоты вершин с которыми связаны как выходящие так и входящие дужки. На рис 8 обозначен допустимый набор потенциалов обеспечивающий минимум по тенциальной функции и соответствующие этому набору высоты. Вычеркивание не максимальных вершин и положительных дужек и вычеркивание по остальным причинам приводит к тому, что в каждой клетке остается единственная вершина, совокупность которых образует допустимый вариант. Этот допустимый вариант, качество которого является оптимальным выделен более толстыми линиями на рис 9.

*Доказательство теоремы 3.* В процессе выполнения одной итерации алгоритма происходит стирание меток на ранее помеченных вершинах, которое в определенных случаях сопровождается нанесением меток на других вершинах. Пусть в некоторый момент времени произошло стирание метки с вершины  $\sigma$  и нанесение метки

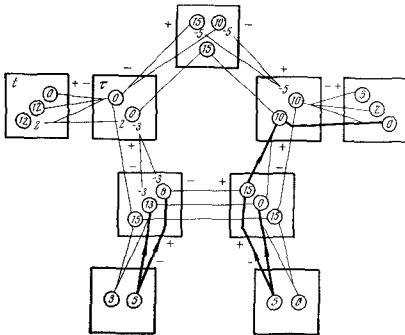


Рис. 7

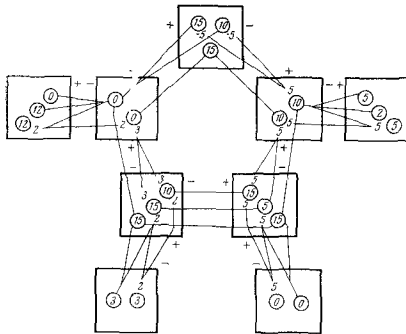


Рис. 8

на некоторой вершине  $\sigma'$ , соединенной вычеркнутой дужкой с вершиной  $\sigma$ . В таком случае будем говорить, что вершины  $\sigma$  и  $\sigma'$  соединены направленной дужкой, выходящей из  $\sigma$  в  $\sigma'$ . Совокупность вершин, хотя бы один раз отмеченных в процессе выполнения одной итерации алгоритма, и введенная совокупность направленных дужек образует граф, который будем называть графом движения меток. Справедливо утверждение, что граф движения меток не содержит циклов. Действительно, из того факта, что из вершины  $\sigma$  выходит дужка движения метки в вершину  $\sigma'$ , следует, что вершина  $\sigma$  вычеркнута по причине  $i(\sigma')$ , следовательно, в момент вычеркивания вершины  $\sigma$  вершина  $\sigma'$  уже была вычеркнута. Рассуждая далее, можно показать, что для любого пути  $\sigma_1, \sigma_2, \dots, \sigma_j$  на графе движения метки в момент вычеркивания вершины  $\sigma_i$  любая вершина  $\sigma_j$  ( $j > i$ ) уже была вычеркнута. Наличие же циклов на графе движения метки означало бы существование пути, в который некоторая вершина входит дважды. Отсюда бы следовало, что в момент ее вычеркивания она была уже вычеркнута. В силу алгоритма это невозможно, так как операция вычеркивания вершин применяется лишь к невычеркнутым вершинам.

Сколько угодно долгое продолжение процесса стирания и нанесения меток неизбежно привело бы к возникновению циклов на графе движения меток. Поэтому доказательство отсутствия таких циклов завершает доказательство теоремы.



Выражение в квадратных скобках содержит не более чем  $\Psi - 1$  слагаемых, каждое из которых не превосходит  $\Psi^{\sigma}$ . Следовательно, абсолютная величина  $\Delta\varphi_{\sigma}^{t(\sigma)}$  не превосходит

$$(\Psi - 1) \Psi^{\tau_{\sigma} + 1} \leq \Psi^{\tau(\sigma)}$$

Вполне очевидно, что ранг любой вершины не превосходит числа  $|\bar{\sigma}|$  всех возможных вершин в множестве  $\bar{\sigma}$ . В силу этого имеем окончательно неравенство

$$(28) \quad |\Delta\varphi_{\sigma}^t| < \Psi^{|\bar{\sigma}|}$$

Доказательство леммы закончено

*Лемма 2.* Для каждой итерации алгоритма величина шага  $\alpha$  либо равна нулю, либо ограничена снизу положительным числом, зависящим от грамматики

*Доказательство.* Известно, что величина шага  $\alpha$  выбирается так, чтобы ее значение было максимальным и в то же время удовлетворяло неравенствам (19), (21), (24).

Покажем, что числители неравенств (19), (21), (24) ограничены снизу некоторыми числами, а знаменатели этих неравенств ограничены сверху положительными числами. Действительно, в неравенстве (19) имеет место соотношение

$$(29) \quad \varphi_{\sigma}^{t(\sigma)} - \varphi_{\sigma'}^{t(\sigma)} > p_2,$$

поскольку дробка ( $\sigma$   $\sigma'$ ) вычеркнута. Для числителей неравенств (21) (24) справедливо выражение

$$(30) \quad h_{t(\sigma)} - h(\sigma) > p_1,$$

так как вершина  $\sigma$  вычеркнута по нулевой причине.

Согласно выражению (15)

$$\max_{\sigma \in \bar{\sigma}^*} \left( \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} - \sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} \right) = 1,$$

а согласно доказательству леммы 1  $|\Delta\varphi_{\sigma}^{t(\sigma)}| < \Psi^{|\bar{\sigma}|}$ , поэтому для знаменателя неравенства (24) справедливо соотношение

$$(31) \quad \max_{\sigma \in \bar{\sigma}^*} \left( \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} - \sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} \right) - \left( \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} - \sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} \right) < < |M(t(\sigma))| \Psi^{|\bar{\sigma}| + 1} \leq \Psi^{|\bar{\sigma}| + 1}$$

Из леммы 1 следует, что для знаменателя в неравенстве (19) справедливо условие

$$(32) \quad \Delta\varphi_{\sigma}^{t(\sigma)} - \Delta\varphi_{\sigma'}^{t(\sigma)} < 2\Psi^{|\bar{\sigma}|},$$

а в неравенстве (21)

$$(33) \quad \sum_{\tau \in M^+(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} - \sum_{\tau \in M^-(t(\sigma))} \Delta\varphi_{\sigma}^{\tau} < |M(t(\sigma))| \Psi^{|\bar{\sigma}|} \leq \Psi^{|\bar{\sigma}| + 1}$$

Учитывая неравенства (29)–(33) и выражения (19) (21), (24), приходим к выводу, что величина шага  $\alpha$  удовлетворяет следующему условию

$$(34) \quad \alpha > \alpha_{\min},$$

где

$$\alpha_{\min} = \min \left( \frac{p_1}{\Psi^{|\bar{\sigma}| + 1}}, \frac{p_2}{2\Psi^{|\bar{\sigma}|}} \right)$$

*Доказательство теоремы 4.* Описанный алгоритм таков, что на каждой итерации алгоритма величина  $\max_{\sigma \in \bar{\sigma}(t)} h(\sigma)$ , где  $t^0$  – нуль клетки, уменьшается на величину,

равную шагу  $\alpha$ , и ни одна из величин  $\max_{\sigma \in \bar{\sigma}(t)} h(\sigma)$  где  $t \neq t^0$  не увеличивается

Следовательно, на каждой итерации алгоритма величина потенциальной функции  $K(\varphi, x) = \sum_{t \in T} \max_{\sigma \in \bar{\sigma}(t)} h(\sigma)$  уменьшается на величину, не меньшую, чем  $\alpha_{\min}$ , которая в силу леммы 2 положительна

К началу первой итерации как следует из описания алгоритма, значение потенциальной функции равно величине  $\sum_{t \in T_0} \max_{v \in L(t)} f(v, x)$  которая по условию теоремы не равна  $\infty$

В силу сказанного неограниченное количество итераций алгоритма привело бы к сколь угодно малым значениям потенциальной функции. Однако это невозможно в силу доказанного в работе [7] утверждения о том, что значение потенциальной функции на множестве допустимых наборов  $\Phi$  ограничено снизу величиной

$$\sum_{t \in T_0} f(v(t), x) \quad \text{где } v - \text{любое изображение допустимое в грамматике } G$$

По условию теоремы грамматика  $G$  непротиворечива следовательно потенциальная функция ограничена снизу Теорема доказана

*Доказательство теоремы 5* Если  $\hat{v}$  - допустимое изображение в грамматике  $G''(\hat{\varphi})$  то, во первых, оно является допустимым и в грамматике  $G$ , и во вторых, существует вариант  $z = (\hat{v}, \hat{s})$ , допустимый в грамматике  $G''(\hat{\varphi})$ . Совокупность вершин входящих в этот вариант, обозначим через  $B\sigma(z)$  а совокупность дужек - через  $B\sigma(\hat{z})$

Для каждой вершины  $\hat{\sigma}$  входящей в вариант  $z$ , справедливо соотношение

$$\max_{\sigma \in \hat{L}(\hat{\sigma})} h(\sigma) - h(\hat{\sigma}) \leq r_1,$$

поскольку если это условие не выполняется для некоторой вершины  $\sigma'$  то эта вершина является вычеркнутой а следовательно не входит в вариант  $z$  т.е.  $\sigma' \notin B\sigma(z)$ . Тогда для всего множества вершин  $B\sigma(\hat{z})$  очевидно неравенство

$$(35) \quad \sum_{t \in T} h_t(\sigma) - \sum_{\sigma \in B\sigma(\hat{z})} h(\sigma) \leq r_1 |T|$$

Из определений высоты вершины (6) (7) следует

$$(36) \quad \sum_{\sigma \in B\sigma(\hat{z})} h(\sigma) = \sum_{t \in T} f(\hat{v}(t), x) - \sum_{\sigma \in B\sigma(\hat{z})} \Delta(\sigma)$$

Справедливо равенство

$$(37) \quad \sum_{\sigma \in B\sigma(\hat{z})} \Delta(\sigma) + \sum_{(\sigma, \sigma') \in B\sigma\sigma(\hat{z})} \delta(\sigma, \sigma') = 0$$

так как любой потенциал  $\varphi_{\sigma'}$  либо входит в обе суммы в этом выражении (если  $\sigma \in B\sigma(\hat{z})$ ), либо не входит ни в одну из сумм (если  $\sigma \notin B\sigma(\hat{z})$ ). При этом если в одну из этих сумм тот или иной потенциал входит со знаком плюс, то в другую сумму он обязательно входит со знаком минус.

Вторая сумма в выражении (37), в силу допустимости набора  $\hat{v}$  и условий вычеркивания дужек является неотрицательной и ограниченной сверху т.е.

$$(38) \quad 0 \leq \sum_{(\sigma, \sigma') \in B\sigma\sigma(\hat{z})} \delta(\sigma, \sigma') \leq r_2 |B\sigma\sigma(\hat{z})| \leq r_2 |T| \Psi / 2$$

поскольку для напряжения  $\delta(\sigma, \sigma')$  на невычеркнутой дужке справедливо условие  $0 \leq \delta(\sigma, \sigma') \leq r_2$

а количество дужек, входящих в множество  $B\sigma\sigma(\hat{z})$ , меньше величин  $|T| \Psi / 2$  или равно ей.

Учитывая равенство (37) и неравенство (38), выражение (36) перепишем в виде

$$(39) \quad \sum_{\sigma \in B\sigma(\hat{z})} h(\sigma) - \sum_{t \in T} f(v(t), x) + \sum_{(\sigma, \sigma') \in B\sigma\sigma(\hat{z})} \delta(\sigma, \sigma') \leq \sum_{t \in T} f(v(t), x) + r_2 |T| \Psi / 2$$

Из неравенств (35), (39) следует, что

$$(40) \quad \sum_{t \in I} \max_{\sigma \in \bar{\sigma}(t(\sigma))} h(\sigma) - \sum_{t \in I} \int v(t, x) \leq r_1 |T| + r_2 |T| \Psi / 2.$$

В работе [7] было доказано утверждение, что функция сходства  $F(v^*, x)$  не превосходит значения потенциальной функции, т.е. что  $K(\Phi, x) - F(v^*, x) \geq 0$  при любом допустимом наборе  $\Phi$  и любом допустимом изображении  $v$ . Вычитая это последнее неравенство из (40), получаем выражение  $F(v^*, x) - F(\hat{v}, x) \leq r_1 |T| + r_2 |T| \Psi / 2$ , которое завершает доказательство.

Поступила в редакцию  
27 января 1976 г.

#### ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. Вапник В. П., Червонович А. Я. Теория распознавания образов. «Наука», 1974.
2. Oja P. A. Mosaic Grammars. University of Pennsylvania, 1973.
3. Sromoney R. On equal matrix languages. Inform. Control., vol. 14, No. 2, pp. 435–454, 1969.
4. Sromoney G., Sromoney R., Krithivasan K. Abstract families of matrices and picture languages. Computer Graphics and Image Processing, vol. 4, No. 3, pp. 284–307, 1972.
5. Pfaltz J. L. Web Grammars and Picture Description. Computer Graphics and Image Processing, vol. 1, No. 2, pp. 193–220, 1972.
6. Ковалевский В. А. Распознавание образов: эвристика или наука? Изд-во Ин-та кибернетики АН УССР, Киев, 1970.
7. Шлесингер М. И. Синтаксический анализ двумерных зрительных сигналов в условиях помех. Кибернетика, № 4, стр. 78–109, 1976.

#### TWO-DIMENSIONAL PROGRAMMING IN IMAGE ANALYSIS PROBLEMS

V. K. KOVAL', M. I. SHLESINGER

An algorithm of analysis of visual images is described and analyzed where special - purpose grammars are used which generate images rather than sequences and are referred to as two - dimensional grammars.