

УДК 677.024.1: 519.15

РАСЧЕТ ЗАПРАВОЧНОГО РИСУНКА ПО ЗАДАННОМУ ТЕХНОЛОГИЧЕСКИ ДОПУСТИМОМУ ТОЧЕЧНОМУ ИЗОБРАЖЕНИЮ

Г.И. БОРЗУНОВ, А.Е. ВОЙНОВ

(Московский государственный текстильный университет им. А.Н. Косыгина)

Ранее в [1..3] были рассмотрены математические модели и алгоритмы, обеспечивающие последовательное получение для произвольного точечного рисунка сначала потенциально допустимого изображения и, наконец, технологически допустимого изображения.

Для технологически допустимого изображения выполняются необходимые условия существования однослойного ткацкого переплетения, которое в сочетании с допустимыми раппортами цвета по основе и утку обеспечивает заданный точечный рисунок.

В настоящей работе предлагается оригинальный алгоритм расчета рационального заправочного рисунка по заданному технологически допустимому изображению.

Ниже приводится псевдокод указанного

алгоритма, в описании которого используется традиционная математическая нотация и элементы языка программирования СИ.

Исходными данными для расчета заправочного рисунка являются: 1) матрица смежности $A[VH][VH]$ и массив меток вершин $W[V][H]$ графа структуры технологически допустимого изображения (G), 2) массив $C[V][H]$, в котором хранятся коды цветов точек технологически допустимого изображения.

Здесь и ниже V – число точек в столбце технологически допустимого изображения, H – число точек в строке технологически допустимого изображения, а значение VH равно произведению: $VH=V \cdot H$. Кроме того задаются нижние и верхние границы для количества основных и уточных перекрытий вдоль нитей основы и утка.

Эти ограничения используются функцией `interlace`, которая при вызове `interlace(W[ii][jj], 1)` возвращает истинное значение, если в матрице переплетения перекрытие с координатами (ii, jj) может быть основным. При вызове `interlace(W[ii][jj], 2)` истинное значение функции означает допустимость уточного перекрытия с теми же координатами. Функция `Adj(A, x, 0)` возвращает множество вершин, принадлежащих окрестности вершины x и соединенные с вершиной x ребрами с весом 0 , а при вызове `Adj(A, x, 1)` функция `j` возвра-

щает вершины окрестности x , но соединенные с вершиной x ребрами с весом 1 . Функция `iiW(x)` возвращает значения первого индекса ii вершины x в массиве вершин `W[ii][jj]`, а функция `jjW(x)` – второго индекса jj вершины x в том же массиве.

В приведенном ниже описании алгоритма отсутствуют определения указанных выше функций. Эти алгоритмы будут рассмотрены в дальнейшем.

Расчет заправочного рисунка реализуется выполнением следующих шагов (1-7):

1. Восстановление рабочих данных

```
/* Восстановление меток вершин графа G */ for(i=0; i< V;i++) for(j=0;j< H ; j++)
W[i][j]=null; /* Освобождение стека помеченных вершин графа G */ while (StMark – непуст)
x=Pop (StMark); /* Установка начальных значений параметров управления циклом поиска не-
помеченной вершины графа G */ jj=0; ii= V –1; перейти к следующему пункту алгоритма
(п.2).
```

2. Поиск непомеченной (свободной) вершины графа структуры технологически допустимого изображения

```
while(jj< H) { while(ii>=0) { if (W[ii][ jj]== null) {/* найдена непомеченная вершина */
if(interlace(W[ii][ jj],1)){/* центр основной волны */ W[ii][ jj]=-1; Push (StMark, W[ii][ jj]);
while (StMark1 – непуст) x=Pop (StMark1);. Push (StMark1, W[ii][ jj]); перейти к построению
уточного фронта (п. 3); } /* Конец оператора if(interlace(W[ii][ jj], 1)) */ elseif(interlace(W[ii][
jj], 2)){/* центр уточной волны */ W[ii][ jj]=-2; Push (StMark, W[ii][ jj]); while (StMark0 – не-
пуст) x=Pop (StMark0); Push (StMark0, W[ii][ jj]); перейти к построению основного фронта
(п. 4); } /* Конец оператора elseif(interlace(x, 2) */ { перейти к откату волны (п. 5); } } /* Ко-
нец оператора if (W[ii][ jj]== null) */
}/* Конец цикла по строкам матрицы W (while(ii>=0)) */
ii= V -1; jj++;}/* Конец цикла по столбцам матрицы W */
перейти к формированию заправочного рисунка (п. 6).
```

3. Построение уточного фронта

```
while (StMark0 – непуст) x=Pop (StMark0);
while(стек StMark1- непуст) {x=Pop(StMark1); for( $\forall W[ii][jj] \in Adj(A, x, 1)$ ) if((W[ii][jj]==1)
|| (W[ii][jj]==-1)) {перейти к откату (п. 5)} if((W[ii][jj]==null) && interlace(W[ii][jj], 2)) {
W[ii][jj]=2; Push (StMark0, W[ii][jj]); Push(StMark,W[ii][jj]);} else {перейти к откату (п. 5)} } /*
Конец оператора for( $\forall W[ii][jj] \in Adj(A, x, 1)$ ) *//* Конец оператора while(стек StMark1 - не-
пуст) */
```

```
if(StMark0 – пуст) {перейти к поиску непомеченной (свободной) вершины графа структу-
ры технологически допустимого изображения (п. 2);} else {Перейти к построению уточного
фронта (п. 4)}
```

4. Построение основного фронта

```
while (StMark1 – непуст) x=Pop (StMark1);
while(стек StMark0 - непуст) { x=Pop(StMark0); for( $\forall W[ii][jj] \in Adj(A, x, 0)$ ) {
if((W[ii][jj]==2) || (W[ii][jj]==-2)) {перейти к откату (п. 5)} if((W[ii][jj]==null) && inter-
lace(W[ii][jj], 1)) { W[ii][jj]=1; Push (StMark1, W[ii][jj]); Push (StMark, W[ii][jj]); } else {перей-
ти к откату (п. 5)} }/* Конец оператора for( $\forall W[ii][jj] \in Adj(A, x, 0)$ ) */ /* Конец оператора
while(стек StMark0 - непуст) */
```

```
if(StMark1 – пуст) {перейти к поиску непомеченной (свободной) вершины графа структу-
```

ры технологически допустимого изображения (п. 2)} else {Перейти к построению уточного фронта (п. 3)}

5. Откат назад

```
while (стек StMark- непуст) { x=Pop(StMark); ii=iiW(x); jj=jjW(x); if((W[ii][jj]==-1)&& (interlace(W[ii][jj], 2))) { /*Установить центр уточной волны */ W[ii][jj]=-2; Push (StMark, W[ii][jj]); Push (StMark0, W[ii][jj]); перейти к построению основного фронта (п. 4);} W[ii][jj]=null;} перейти к завершению работы алгоритма (п.7).
```

6. Расчет заправочного рисунка

```
/*Установка начальных значений раппортов цвета по утку ColV и по основе ColH ( -1 обозначает, что цвет не выбран) */
```

```
for(i=0;i< V;i++) {ColV[i]=-1; } for(i=0;i< H; i++) ColH[i]=-1; }
```

```
/* Расчет раппортов цвета по основе и утку */
```

```
for(i=0; i< V;i++) { for(j=0;j< H ; j++) {if((W[i][j]==1)|| (W[i][j]==-1)) if(ColH[j]==-1) ColH[j]=C[i][j]; else {if(ColV[i]==-1) ColV[i]= C[i][j];} } /* Конец цикла for(j=0;j< H ; j++) */ } /* Конец цикла for(i=0;i< V;i++) */
```

```
/* Структура Z объединяет части заправочного рисунка */
```

```
Z={W, ColH, ColV};
```

```
Push (Queuez, Z);
```

```
перейти к откату назад (п. 5).
```

7. Завершение работы алгоритма

```
if (очередь Queuez – пуста) {Вывести сообщение об отсутствии решения; стоп.} while (очередь Queuez – непуста) { Z=Pop(Queuez); Вывести Z;} Вывести сообщения о завершении работы; стоп.
```

ВЫВОДЫ

1. Рассчитанный по предлагаемому в данной работе алгоритму заправочный рисунок обеспечивает выработку исходного точечного рисунка с близкими к минимально возможным искажениями.

2. Структура алгоритма позволяет при его реализации эффективно использовать стандартные объектно-ориентированные библиотеки языка программирования СИ++.

ЛИТЕРАТУРА

1. Борзунов Г.И. // Изв. вузов. Технология текстильной промышленности. –1981, №5. С.42...45.
2. Борзунов Г.И. // Изв. вузов. Технология текстильной промышленности. – 1985, №5. С. 35...41.
3. Борзунов Г.И. // Изв. вузов. Технология текстильной промышленности. – 2006, №5. С.121...123.

Рекомендована кафедрой информационных технологий и вычислительной техники. Поступила 29.11.06.