

УДК 677.024.1: 519.15

**МАТЕМАТИЧЕСКАЯ МОДЕЛЬ КОРРЕКЦИИ
ЦВЕТНЫХ ТОЧЕЧНЫХ РИСУНКОВ**

Г.И. БОРЗУНОВ

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

В работе [1] была предложена математическая модель тканых узоров, вырабатываемых с использованием однослойных ткацких переплетений. Эта математическая модель представляет собой граф структуры цветного рисунка, вершины которого соответствуют цветным точкам заданного рисунка, а каждое ребро окрашено в один из двух цветов, обозначаемых нулем или единицей.

Указанная математическая модель обеспечивает эффективный анализ возможности выработки заданных цветных рисунков при автоматизированных расчетах заправочных рисунков мелкоузорчатых тканей [2], но потребовалось ее теоретическое развитие для обоснования и реализации эффективного алгоритма коррекции цветных рисунков, выработка которых невозможна с использованием однослойных ткацких переплетений. Это развитие математической модели тканых узоров рассматривается в данной работе.

шинами j и k четырех маршрутов, вид которых показан на рис.1, делает невозможным выработку заданного рисунка с использованием однослойного ткацкого переплетения.

Но легко видеть, что эти 4 маршрута составляют два цикла нечетной длины, проходящих через вершину i и образующих «восьмерку». Один из этих сдвоенных циклов, кроме вершины i , проходит через вершину j , а другой – через вершину k .

Таким образом, для существования ткацкого переплетения, обеспечивающего выработку заданного рисунка, необходимо отсутствие в графе структуры рисунка циклов нечетной длины, проходящих через одну и ту же вершину так, как показано на рис. 1.

Пользуясь терминологией, предложенной в [3, с. 247], такие сдвоенные циклы можно назвать запрещенными фигурами.

В той же работе [3, с.254], рассматриваются два подхода к устранению запрещенных фигур, являющихся обычными циклами нечетной длины: 1) удаление из графа одного из ребер, принадлежащих циклу; 2) расщепление одной из вершин, принадлежащих циклу.

В данном случае при сохранении числа точек в рисунке ограничений на вид используемых переплетений возможно только удаление ребра из графа структуры рисунка. Для удаления ребра из графа достаточно выполнить следующие действия: 1) определить в рисунке две точки, соответствующие вершинам, соединенным ребром, которое подлежит удалению (x_i, x_j); 2)

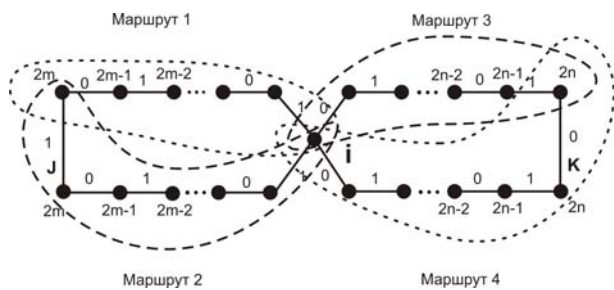


Рис. 1

В [1] показано, что наличие в графе структуры соединяющих вершину i с вер-

определить цвета этих точек $C[x_i]$, $C[x_j]$;
 3) присвоить точке x_i цвет точки x_j ($C[x_i] = C[x_j]$) или, наоборот, присвоить точке x_j цвет точки x_i ($C[x_j] = C[x_i]$).

Следовательно, для устранения запрещенных фигур требуется изменить цвет некоторых точек исходного рисунка. Число таких изменений определяется выбором точек, меняющих свой цвет, а также выбором цветов, в которые эти точки перекрашиваются.

Ниже впервые предлагается алгоритм, который реализует удаление запрещенных фигур из графа структуры цветного рисунка, изменяя цвет минимального числа точек рисунка. Алгоритм представляет собой итеративно повторяющуюся процедуру, которая осуществляется в два этапа.

На первом этапе для каждой вершины определяется, какое количество запрещенных фигур инцидентно этой вершине, и определяется общее число запрещенных фигур в структурном графе. Если запрещенных фигур в графе нет, то алгоритм завершает свою работу.

На втором этапе определяется вершина, окрашивание которой в цвет одной из смежных с нею вершин приводит к уничтожению максимального числа запрещенных фигур. Далее выбранная вершина окрашивается в указанный цвет и осуществ-

ляется итеративный переход к этапу 1.

Описанный выше алгоритм построен по схеме «жадных» алгоритмов и, очевидно, всегда обеспечивает удаление всех запрещенных фигур за конечное число шагов. Как показано в [4 с. 268], такие алгоритмы оказываются достаточно быстрыми и даже, если не гарантируют получение оптимальных решений, то с большой вероятностью позволяют получать «хорошие» решения, всего на несколько процентов отличающиеся от оптимальных решений.

Рассмотрим более подробно определение числа запрещенных фигур, инцидентных некоторой вершине. Эта часть алгоритма основывается на схеме базового алгоритма обхода вершин графа «в ширину» [4, с.218]. Эта схема была переработана с учетом особенностей графа структуры рисунка.

Ниже приводится псевдокод этого алгоритма (Bsf), в описании которого использована традиционная математическая нотация и обозначения, принятые в языке СИ:

(* G – граф структуры, v – вершина, для которой определяется число инцидентных ей запрещенных фигур, Queue0 и Queue1 – очереди активных вершин, помеченных соответственно нулем и единицей*)

```

Bsf(G, v)
/*Определение числа левых частей запрещенных фигур (см. рис. 1)*/
ResetMark(G); Visit(v,1); Push(Queue1,v);
While(очередь Queue0 или Queue1 - непушта)
{ /*Начало цикла While(очередь Queue0 или Queue1 - непушта)*/
While(очередь Queue1 - непушта){w=Pop(Queue1);for(i ∈ Adj(G, w, 1))
{ if(i – не помечена) { Visit(G, i, 0); Push(Queue0, i); } } }
While(очередь Queue0 - непушта){w=Pop(Queue0);for(i ∈ Adj(G, w, 0))
{ if(i – не помечена) { Visit(G, i, 1); Push(Queue1, i); } } }
} /*Конец цикла While(очередь Queue0 или Queue1 - непушта)*/
/*Определение числа правых частей запрещенных фигур (см. рис. 1)*/
ResetMark(G); Visit(v,0); Push(Queue0,v);
While(очередь Queue0 или Queue1 - непушта)
{ /*Начало цикла While(очередь Queue0 или Queue1 - непушта)*/
While(очередь Queue0 - непушта){w=Pop(Queue0);for(i ∈ Adj(G, w, 0))
{ if(i – не помечена) { Visit(G, i, 1); Push(Queue1, i); } } }
While(очередь Queue1 - непушта){w=Pop(Queue1);for(i ∈ Adj(G, w, 1))
{ if(i – не помечена) { Visit(G, i, 0); Push(Queue0, i); } } }
} /*Конец цикла While(очередь Queue0 или Queue1 - непушта)*/

```

В приведенном выше псевдокоде Bsf используются функции, выполняющие следующие действия: ResetMark(G) – восстановление меток вершин графа G; Push(Queue0,v); Push(Queue1,v), Pop(Queue1), Pop(Queue0) – помещение вершины v в очередь, заданную аргументом, или извлечение вершины из очереди, заданной аргументом; Adj – генерация окрестности вершины w, образуемой ребрами цвета 0 или 1; Visit(v,0), Visit(v,1) – моделирование посещения не помеченных вершин, маркировку вершин соответственно нулем или единицей, подсчет фрагментов запрещенных фигур, выявленных на данном шаге.

ВЫВОДЫ

1. Впервые предложенная математическая модель коррекции точечных цветных рисунков обеспечивает автоматизацию этой процедуры и может служить основой для программной реализации соответствующего алгоритма.

2. Результаты коррекции рисунка оцениваются экспертным путем, что делает

необходимой визуализацию этих результатов на экране монитора с возможностью их изменения.

3. В результате итеративно повторяющихся процедур автоматической коррекции рисунка и интерактивного изменения этих результатов, достигается компромисс между точностью воспроизведения рисунка и технологическими ограничениями качества.

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. Горбатов В.А. Основы дискретной математики: Учебное пособие для студентов вузов. – М.: Высшая школа, 1986.
4. Ахо А.В., Хопкрофт Дж., Ульман Дж.Д. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2001.

Рекомендована кафедрой информационных технологий и вычислительной техники. Поступила 01.06.06.