Оптимизация времени отображения векторных графических изображений большого размера

Кетков Ю.Л., Кирьянов С.К, Нижегородский университет

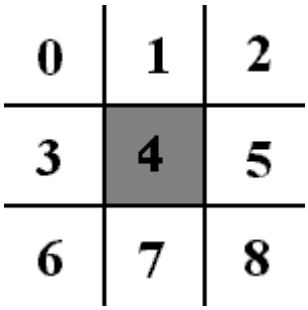
В докладе рассматриваются быстрые алгоритмы фрагментации векторных графических файлов большого размера. Процедура формирования индексных массивов, определяющих принадлежность графических примитивов тем или иным прямоугольным фрагментам изображения, практически не замедляет процесс первоначального отображения содержимого файла целиком. Последующий просмотр любых фрагментов документа с требуемым увеличением сопряжен с задержкой не более 1-2 сек.

Визуальный контроль цифровых электронных карт при подготовке их к изданию связан со скрупулезным просмотром на экране дисплея графических файлов, размер которых достигает нескольких десятков (в векторном формате) и даже сотен (в растровом формате) мегабайт. Дефекты изображения, не допустимые соответствующими руководствами, приходится тем или иным способом исправлять. Векторное изображение электронной карты, полученное по соответствующей цифровой базе, отличается большей гибкостью как в плане масштабирования, так и в плане внесения изменений в существующий документ.

В докладе рассматриваются алгоритмы фрагментации векторного изображения, заложенные в системы отображения цифровых карт, разрабатываемые в Научно-исследовательском институте прикладной математики и кибернетики при Нижегородском университете. Конкретный графический формат представления векторного изображения на алгоритмы фрагментации существенного влияния не оказывает. Картографический документ, в частности, может быть сформирован на языке HP-GL, хотя описываемый комплекс программ был ориентирован на некоторый внутренний формат, представленный записями переменной длины. Среди них присутствуют записи, изменяющие текущие параметры пишущего узла, управляющие построением видимых и невидимых отрезков и других графических примитивов, ограничивающие область заливки (штриховки) площадных объектов и т.п.

Задача фрагментации векторного изображения сводится к определению принадлежности очередной графической записи той или иной прямоугольной области, на которые статически или динамически разбит исходный документ. Возможно, что отдельные записи в файле определяют графический объект, принадлежащий сразу нескольким прямоугольникам (например, лес или озеро достаточно большой площади). Процедура выделения в таких данных подобъектов, принадлежащих только одному прямоугольнику, достаточно трудоемка. Гораздо проще причислить такую запись одновременно нескольким индексным массивам, перелагая задачу отсечения ненужных участков на соответствующие функции GDI.

Классификация точек и отрезков прямых. **О**сновная нагрузка в файлах с векторным описанием картографических документов падает на небольшие перемещения пишущего узла вдоль отрезков ломаных линий. Поэтому в первую очередь мы рассмотрим задачу классификации записей, переводящих перо в поднятом или опущенном состоянии в точку с заданными координатами (x,y). Пронумеруем области, на которые плоскость разбивается анализируемым прямоугольником:



Обозначим через `xy` массив с координатами противоположных вершин прямоугольника (`xmin, ymin, xmax, ymax`),. Достаточно эффективная программа определения номера области, которой принадлежит точка `(x,y)` представлена функцией `index` (индексы точек и индексы записей имеют разный смысл):

```
int index(float x, float y, float *xy)
{ register int d=0;
  if(x<xy[0])  goto m; d++;
  if(x<=xy[2]) goto m; d++;
m:
  if(y>xy[3])  return d;
  if(y<xy[1])  return d+6;
  return d+3; }
```

Если очередная запись переводит перо в поднятом состоянии в точку `(x,y)`, не принадлежащую области 4, то индекс этой записи в список индексов прямоугольника не включается. Однако координаты точки надо запомнить в качестве текущей точки CP `(xold,yold)`. Вполне возможно, что следующий видимый отрезок, проведенный из CP, может попасть в прямоугольник или пересечь его.

Если координаты очередной точки принадлежат области 4, то индекс соответствующей записи должен быть включен в массив индексов прямоугольника независимо от того, в каком состоянии перо переводится в эту точку. И координаты CP должны быть заменены новой точкой.

Для анализа записи, переводящей перо в опущенном состоянии в точку с координатами `(xnew,ynew)`, предлагается использовать индексы текущей `(ind_old)` и новой `(ind_new)` точек. Минимизация числа операций на этом этапе достигается за счет использования матрицы `ij` размером 9×9 (см. текст функции `in_out`) Строки и столбцы этой матрицы соответствуют индексам точек начала и конца анализируемого отрезка. Элементы матрицы подобраны таким образом, чтобы свести к минимуму число строк программы:

-1 – анализируемый отрезок не принадлежит прямоугольнику;

0 – проверить пересечение отрезка с прямой `x=xmin;`

1 – проверить пересечение отрезка с прямой `y=ymin;`

2 – проверить пересечение отрезка с прямой `x=xmax;`

3 – проверить пересечение отрезка с прямой `y=ymax;`

4 –проверить пересечение отрезка с прямыми `x=xmin` и `x=xmax;`

5 –. анализируемый отрезок принадлежит прямоугольнику;

Функция `in_out` возвращает значение 1, если индекс записи с новой точкой принадлежит массиву индексов прямоугольника, и 0 – в противном случае.

```
  int in_out(int jold,float xold,float
yold,
             int jnew,float xnew,float
ynew,float *xy)
{ int j;
  float x,y,A,B,C;
  int ij[9][9]={
{-1,-1,-1,-1, 5, 2,-1, 1, 4},
{-1,-1,-1, 0, 5, 2, 3, 5, 3},
{-1,-1,-1, 0, 5,-1, 4, 1,-1},
{-1, 0, 0,-1, 5, 5,-1, 0, 0},
{ 5, 5, 5, 5, 5, 5, 5, 5, 5},
{ 2, 2,-1, 5, 5,-1, 2, 2,-1},
{-1, 0, 4,-1, 5, 2,-1,-1,-1},
{ 1, 5, 1, 0, 5, 2,-1,-1,-1},
{ 4, 3,-1, 0, 5,-1,-1,-1,-1} };
  j=jj[jold][jnew];
  if(j>4) goto m1; //j=5
  if(j<0) goto m0; //j=-1
//находим коэффициенты
//уравнения прямой
  A=ynew-yold;
  B=xold-xnew;
  C=-B*yold-A*xold;
  if(j>3){//j=4
  x=xy[0]; y=-(A*x+C)/B;
  if(xy[1]<=y && y<=xy[3])  goto m1;
  x=xy[2]; y=-(A*y+C)/B;
  if(xy[1]<=y && y<=xy[3]) goto m1;
  goto m0;}
  if(j&0x01) {//j=1 или j=3
    y=xy[j]; x=-(B*y+C)/A;
  if(xy[0]<=x && x<=xy[2]) goto m1; goto
m0; }
//j=0 или j=2
  x=xy[j]; y=-(A*x+C)/B;
  if(xy[1]<=y && y<=xy[3]) goto m1;
m0: return 0;
m1: return 1; }
```

Классификация записи изменения цвета пера. Запись, изменяющая цвет пера, фиксируется в массивах индексов каждого прямоугольника.

Классификация площадных объектов. Записи, представляющие заливаемые односвязные области, содержат счетчик точек контура и координаты вершин соответствующего полигона в том или ином формате представления чисел. Для анализа принадлежности такой записи прямоугольнику определяются параметры габаритного прямоугольника, охватывающего площадной знак. Затем вершины каждого прямоугольника проверяются на принадлежность области 4 другого прямоугольника. Не подпадающие под этот алгоритм варианты пересечения прямоугольников типа "крест" в реальных картографических документах не встречаются.

Применение описанных алгоритмов для фрагментации векторных изображений топографических карт позволило достичь следующих результатов. На ПК с процессором Celeron-733 и оперативной памятью 128 Мбайт отображение содержимого векторного файла объемом порядка 5-6 Мбайт занимало около 20 сек. На просмотр любого фрагмента такого документа без фрагментации изображения тратились те же 20 сек, т.к. приходилось обрабатывать весь файл целиком. После построения индексных массивов для приемлемой прямоугольной сетки время просмотра любого фрагмента сократилось до 1-2 сек. Очевидно, что при оперативной памяти нормального размера графический файл и индексные массивы целесообразно хранить в памяти, а не подкачивать нужную порцию с диска по мере необходимости.

Сведения об авторах:

Кетков Юлий Лазаревич, д.т.н., профессор Нижегородского университета, зав. отделом Научно - исследовательского института прикладной математики и кибернетики:

e-mail – ket@city.ru

раб. тел – (8-831-2)38-99-40

Кирьянов Сергей Константинович, магистр факультета ВМК ННГУ.