

Использование TTF-шрифтов при подготовке издательских оригиналов электронных карт

Кузнецов А.И.
Институт Прикладной Математики и Кибернетики
г. Нижний Новгород, Россия

Аннотация

В докладе описывается технология использования шрифтов TTF в картографической издательской деятельности, которая внедрена в систему подготовки издательских оригиналов географических карт и планов, функционирующую под управлением Windows. Проводится её сравнение с технологией, ранее применявшейся на платформе MS-DOS.

Ключевые слова: Геоинформационные системы, инструментальные средства, векторная графика, HP-GL, TTF, шрифт.

Традиции российских картографов, заложенные еще во времена Петра I, предъявляют очень высокие требования к оформлению карт. Особенно большое внимание при подготовке издательских оригиналов уделяется воспроизведению подписей. Количество картографических шрифтов, выделяющих те или иные объекты, исчисляется десятками. В ранних версиях системы КАРТ-ДОК, разработанной в НИИ ПМК для визуализации и подготовки к изданию топографических карт, использовались шрифты в формате BGI (Borland Graphics Interface), на создание которых ушло порядка двух лет. Их основные недостатки – использование только линейных отрезков для описания контура и отсутствие заливки.

Основные технологические шаги при работе с системой КАРТ-ДОК следующие:

1. ввод карты/плана. На этом этапе получается база данных с семантическим описанием листа карты (цифровая модель карты).
2. преобразование базы данных в графическое описание на языке HP-GL¹.
3. выдача описания, полученного на шаге 2 на устройство вывода (струйный или лазерный плоттер, экран и т.д.).

Наиболее интересным местом в этой схеме является шаг 2. Здесь каждый объект, присутствующий

в базе данных, получает свой графический облик. Этот облик, иногда, отличается от того, который объект имел на исходной карте. Происходит это из-за того, что условные знаки на картах иногда меняются, поэтому карты нужно переиздавать с учётом новых правил. Наиболее часто используемые картографические знаки (т.н. дискретные знаки) собраны в библиотеки знаков, которые и используются при преобразовании в программу на языке HP-GL. Подобные же библиотеки были созданы и для более сложных по своей структуре географических знаков (площадных и линейных). Методика преобразования базы данных в графическое описание свелась к сопоставлению объектам базы графического описания из библиотеки и записи этого описания в результирующий файл.

Ахиллесовой пятой данной технологии на протяжении многих лет было отображение текстовой информации. Со стороны может показаться, что ничего особенного в отображении текста нет. Но это не так. Каждый вид надписи должен быть сделан своим, специально предназначенным для этого шрифтом с соответствующими размерами, весом, стилем и т.д. и т.п. На практике количество шрифтов не так велико. Большинство карт можно нарисовать, используя 26 шрифтов. Тем не менее, разработка даже одного шрифта может занять не один год. Все буквы должны быть соразмерными, иметь один стиль, вес, наклон, унифицированно смотреться на издаваемом документе и т.д.

Начальные версии КАРТ-ДОК'а использовали шрифты, основанные на BGI-шрифтах. Стандарт шрифтов BGI был предложен фирмой Borland в её семействе продуктов Turbo C++/Pascal. Этот формат тоже являлся векторным и получил в своё время широкое распространение в программах, работающих под управлением DOS. Система КАРТ-ДОК имела четыре основных шрифта и таблицу генерации недостающих 22х. Эта таблица содержала информацию о требуемом наклоне, коэффициентах масштабирования и порогах высот. По этой информации специальные подпрограммы деформировали один из базовых шрифтов и получали требуемый.

Такой подход мало устраивал картографов. Дело в том, что на карте имеются как очень маленькие подписи, так и очень крупные, а КАРТ-ДОК воспроизводил их контуры однообразно. В результате на очень крупных надписях давала о себе знать

¹ Язык HP-GL – это язык векторного описания графической информации, разработанный фирмой Hewlett Packard для своих плоттеров. Данный язык стал почти таким же популярным, как и всем известный PostScript[®]. (PostScript стал стандартом печати для многих издательских систем)

векторная сущность шрифтов (некоторые участки букв были заштрихованы отрезками прямых, между которыми появлялись пробелы при большом увеличении), чётко выявлялись отрезки, из которых состоят буквы. На очень мелких надписях появлялось ощущение утолщённости линий в некоторых местах букв. Некоторые шрифты вообще масштабировались неправильно (например, на шрифтах в одну линию при большом увеличении начинали сказываться способы окончания отрезков).

Самым большим недостатком данной технологии представления шрифтов являлось то, что шрифты было очень трудно модифицировать. Существовала одна единственная программа по редактированию таких шрифтов – Borland Font Editor, которая имела очень скудные возможности. Более того, при изменении одного базового шрифта никто не мог поручиться за то, что какие-нибудь из производных в результате этого не окажутся испорченными.

Была предпринята попытка представить каждую букву в виде описания на языке HP-GL (как и другие условные знаки). Однако, данное представление не слишком отличается от того, что уже есть.

В 2002-2005 годы был выполнен перевод системы КАРТ-ДОК на платформу Windows XP. В связи с этим переходом пришлось отказаться от многих DOS-овских программ по работе с BGI-шрифтами. Было решено модифицировать и всю работу с текстами. Самым распространённым стандартом представления шрифтов является TTF (True Type Font). Данный формат получил распространение практически на всех существующих сейчас платформах (UNIX, Windows, Apple). Поэтому и было принято решение пользоваться именно этим форматом. Кроме того, для редактирования TTF-шрифтов существует большое количество графических редакторов, что исключает проблемы с их модификацией.

Отсутствие картографических шрифтов могло очень замедлить внедрение новой технологии, однако на выручку пришла программа сканирования шрифтов ScanFont. Эта программа позволяет читать отсканированные растровые образы букв и строить для них TTF-описание. Поэтому, базовый набор шрифтов был создан всего лишь за несколько дней.

Для получения изображения каждой буквы на языке HP-GL всё-таки требуется её векторное описание. Прикладной интерфейс Windows XP имеет в своём составе функцию GetGlyphOutline, которая позволяет извлекать метрическую информацию о глифе. Глифом в терминологии шрифтов TTF называется изображение буквы (некоторые буквы имеют один глиф, например, латинское и русское «В»). Фактически, функция GetGlyphOutline позволяет получить информацию (т.н. контекст растеризатора), по которой растеризатор Windows получает растровое изображение буквы для её отображения на экране дисплея. Перед тем, как получить контекст

растеризатора, нужно выполнить следующие предварительные шаги:

1. m_LogData=.; // Создание логического шрифта.
2. m_dcDisplay=CreateDC("DISPLAY",NULL,NULL, NULL); // Создание контекста дисплея.
3. m_fAnFont=CreateFontIndirect(&m_LogData); // Создание шрифта в соответствии с логической информацией.
4. m_fOldFont=(HFONT)SelectObject(m_dcDisplay, m_fAnFont); // Выбор шрифта в контекст устройства.
5. GetOutlineTextMetrics (m_dcDisplay, sizeof(m_Outline), &m_Outline); // Получение метрической информации о шрифте в целом.
6. m_nKernSize=GetKerningPairs(m_dcDisplay, 0, NULL); // Получение информации о размере кернинга.
7. m_pKerningData=new KERNINGPAIR[m_nKernSize];
8. GetKerningPairs(m_dcDisplay, m_nKernSize, m_pKerningData); // Получение информации о кернинге в шрифте.

Эти шаги выполняются каждый раз при смене шрифта (они занимают немного времени). Получение контекста растеризатора выполняется следующим образом:

1. mode=GGO_NATIVE; // Родной формат растеризатора.
2. bufsize=GetGlyphOutline(m_dcDisplay, chgl, mode, &m_gmMetrics, 0, NULL, &trans); // Получение информации о размере данных.
3. buffer=new BYTE[bufsize];
4. bufsize=GetGlyphOutline(m_dcDisplay, chgl, mode, &m_gmMetrics, bufsize, buffer, &trans); // Получение данных растеризатора.

После выполнения этих строк в участке памяти buffer будут располагаться данные растеризатора. Здесь кроется очень большой подводный камень: чем меньше размер заказанной буквы (чем она дальше от точки наблюдения), тем грубее получаемые данные растеризатора. Чтобы это избежать, нужно всегда запрашивать буквы очень большого размера (например, мы используем размер 600 точек).

Для разбора данных растеризатора была написана довольно большая подпрограмма, которая возвращает массив полигонов² с дырками. Далее к полученному массиву полигонов применяется аффинное преобразование, чтобы придать букве требуемые наклон и высоту.

В результате, весь механизм рисования однострочной надписи text[] можно представить следующим простым алгоритмом:

1. i=0; CP=A; // установили начальную точку.

² Полигон – конечная область, ограниченная замкнутой ломаной.

2. Получить контекст растеризатора для буквы `text[i]`.
3. Преобразовать метрику буквы в соответствии с требуемой высотой подписи `h`, начальной точкой `CP` и вектором наклона надписи `v`.
4. Отобразить метрику.
5. $CP = CP + v * (\text{длина_буквы} - \text{кернинг}(i, i+1))$.
6. Если не вся строка отображена, $i = i + 1$ и перейти на 2.

Перечислим достоинства и недостатки данной технологии. Теперь каждая буква представлена несколькими залитыми полигонами с дырками, что позволяет её произвольно масштабировать с сохранением всех пропорций. Кроме того, появилось ещё одно заметное улучшение качества – поддержка кернинга. Кернинг – это субтехнология TTF, которая позволяет устранять моноширинность шрифтов, вносит в их описание информацию о взаимодействии между отдельными буквами. Например, в сочетании «ГА» можно заметить, что буква «А» въехала в знакоместо буквы «Г». В информации о кернинге для такого сочетания указывается, насколько очередную букву можно сместить внутрь знакоместа предыдущей буквы. Раньше, когда кернинг не использовался, в таких случаях в надписи можно было заметить пробелы. Эти пробелы являются чисто зрительными, т.к. знакоместа букв были состыкованы по границам (особенно заметны пробелы вокруг цифры «1»).

За качество, как известно, приходится платить. В данном случае так и произошло. После введения TTF-шрифтов размеры выходных файлов на языке HP-GL выросли в три раза. Объясняется это просто. Залитый контур с дырками представляется на HP-GL в виде следующей программы:

PM0;

<описание основного контура>

PM1;

<дырка1>

...

PM1;

<дырка N>

PM2;

FP;

Такие вставки заменили собой места, где раньше стояли простые команды обводки контуров.

На производительность переход на новую технологию не сказался никак.

После внедрения данного способа отображения шрифтов выяснилось, что эта же программа (без каких-либо изменений) отлично работает и для многих других типов шрифтов, поддерживаемых WindowsXP (OTF, PFA, PFB...).

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 05-01-00590).

Литература

[1] Кетков Ю.Л., Лютова Н.В., Тафорина Н.М., Урусова Л.А., Инструментальный комплекс для формирования библиотек картографических шрифтов и дискретных условных знаков. // 2-я Всерос. конф. "Распознавание образов и анализ изображений: новые инф. технологии", ч.4 - Ульяновск, 1995.

Об авторе

Кузнецов Александр Иванович – инженер-программист НИИ ПМК. E-mail: anubis1@rambler.ru.

Using of the TTF in creating of geographical map publishing form

Abstract

This report describes a new technology of using of TTF-fonts in the map publishing business. It contains basic description of glyph retrieval process in Windows XP environment and comparison of new and old technologies for font representation. The old technology was being used by KART-DOK system on DOS platform. KART-DOK is a complex system for creating of publishing forms of maps and plans

About the author

Kusnetsov Alexander Ivanovitch is a programmer of The Institute for Applied Mathematics and Cybernetics, Nizhny Novgorod, Russian Federation.