УДК 681.327.2 /.7

Д.А. Дровнинов, К.Н. Марков, М.Г. Суханов

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ ПРОСТРАНСТВЕННЫХ ДАННЫХ GDAL/OGR ДЛЯ РАБОТЫ С ОТЕЧЕСТВЕННЫМИ ФОРМАТАМИ ДАННЫХ

Введение

Разработка ГИС сопряжена с рядом сложностей. Одна из которых связана с форматами хранения данных. Данную проблему обычно решают двумя способами: используют стандартные (широко распространенные) форматы данных или создают собственные. Выбор между этими подходами не так однозначен. Применение стандартных форматов обеспечивает совместимость с «мировым сообществом», но при этом могут возникать сложности с тем, что зачастую такие форматы не поддерживают необходимый для программы функционал или не удовлетворяют требованиям по скорости и удобству доступа. Наряду с этим существуют проблемы, связанные с реализацией драйвера доступа и отсутствием документации к формату, или она есть, но недостаточно подробна. Также некоторые форматы имеют ограничения, связанные с лицензией к ним.

Создание собственного формата хранения данных позволяет реализовать все самые смелые пожелания разработчиков. Но для взаимодействия с другими ГИС необходимо будет разрабатывать механизмы экспорта и импорта.

Оба подхода имеют право на существование. Для ГИС, ориентированных на работу с данными (задачи моделирования или другие расчетные задачи) целесообразно использовать свой формат, так как это позволит оптимизировать доступ к данным, тем самым достичь максимальной производительности. Процедура импорта в данном случае занимает малое время из общего объема работы приложения, хотя и сопряжена с некоторыми неудобствами. В ГИС, ориентированных на задачи оформления карт применяют оба подхода. ESRI ArcGIS использует в качестве базового формата собственную разработку – ESRI Shapefile, который уже стал отраслевым. Например, QuantumGIS – использует почти любые форматы данных.

При выборе формата хранения данных для «презентационных» ГИС важной является задача поддержки актуальности данных. Если они меняются достаточно быстро и поступают из независимых источников в общераспространенных или транспортных форматах, например, в формате SXF – ГИС Панорама, то использование собственных форматов становится затруднительно. В таком случае необходимо применять специальную библиотеку, осуществляющую стандартный доступ сразу ко всем форматам или реализовывать необходимые форматы самостоятельно.

Реализация доступа к данным унифицированным способом посредством специальных библиотек приводит к потерям в скорости доступа из-за невозможности обеспечить механизм чтения, учитывающий их особенности, и уникальные возможности форматов.

Самостоятельная разработка необходимых драйверов позволяет учитывать их уникальные возможности, но из-за большого количества всех распространенных на сегодняшний день форматов реализовать поддержку достаточно трудоемко. Если разработка осуществляется собственными силами, то обычно ограничиваются поддержкой только самых распространенных.

При создании многофункционального геоинформационного сервера (разработка ФГУП ГНЦ РФ ВНИИгеосистем) одним из требований было обеспечение доступа к максимально большому числу форматов. Разработка такой объемной задачи собственными силами была признана нерациональной, в результате чего было принято решение использовать свободно распространяемый программный продукт, созданный международным сообществом, – библиотеку стандартного доступа к растровым и векторным форматам данных – GDAL/OGR.

Библиотека GDAL/OGR (http://www.gdal.org) разрабатывается Open Source Geospatial Foundation. Она обеспечивает универсальный доступ более чем к 100 растровым (библиотека GDAL) и векторным (OGR, входящая в GDAL) форматам, например, к файловым форматам (GeoTiff, ArcInfo grid, Golden Software grid, ESRI Shapefile, Autocad DWG, MapInfo, GML и другие), базам данных (Oracle, PostGIS, ArcSDE, Microsoft SQL Server), веб-сервисам (WMS, WFS и другим).

Библиотека GDAL/OGR уже нашла свое применение во многих промышленных проектах, таких как ESRI ArcGis, GoogleEarth, UMN MapServer, QuantumGIS и многих других. Программный интерфейс доступен для следующих языков программирования: C/C++, Perl, Python, VB6, Ruby, Java, C#/.Net. GDAL/OGR реализована под различные операционные системы: MS Windows, Lunux, FreeBSD.

Кроме предоставления доступа к данным, библиотека также содержит дополнительные инструменты по работе с ними. Поддерживается функция перепроецирования как растровых, так и векторных данных. Работа с проекциями осуществляется посредством библиотеки PROJ.4. Для векторных данных поддерживаются операции с геометрией объектов (слияние, вычитание и т.д.), логические операции (проверка на идентичность, включает ли один полигон другой и т.д.). Поддерживается возможность интеграции с библиотекой операций над векторной графикой (GEOS) и растеризации векторных данных.

К основным возможностям библиотеки GDAL по работе с растровыми данными относятся:

- обеспечение работы с многослойными растровыми данными;
- получение информации о геометрии (линейные размеры, шаг сетки), маркер отсутствия значений, данные о проекции;
- получение данных из слоя любого типа (байт, целое, дробное и другие);
- возможность запроса данных прямоугольной областью из любого места слоя;
- возможность работы с пирамидой слоев;
- возможность проведения операций преобразования форматов данных.

Основными возможностями библиотеки OGR для работы с векторными данными являются:

- обеспечение работы с многослойными векторными данными;
- получение информации о файле и о слоях: охватывающая рамка, число слоев, тип слоев, число полей в таблице атрибутов и др.;
- обеспечение доступа к объектам слоя последовательно или в случае поддержки в драйвере формата произвольного доступа к данным – по индексу;
- доступ к полям таблицы атрибутов;
- возможность использования пространственного и атрибутивного фильтра;
- возможность проведения операций преобразования форматов данных.

Еще одним достоинством библиотеки GDAL/OGR является то, что она предоставляет возможность расширять свой функционал путем добавления собственных драйверов доступа к форматам данных.

Во ФГУП ГНЦ РФ ВНИИгеосистем на основе этой библиотеки разработаны драйверы для формата ГИС ИНТЕГРО (TOC4) и российского формата обмена картографическими данными – ГИС Панорама (SXF), о которых пойдет речь ниже.

Модель данных GDAL

Модель данных GDAL [1] описывает программное представление структуры растровых данных (рис. 1). При работе используются следующие понятия:

Источник данных (представляемый классом GDALDataset) состоит из связанных растровых каналов (слоев), а также некоторой дополнительной общей информации. Общей информацией являются

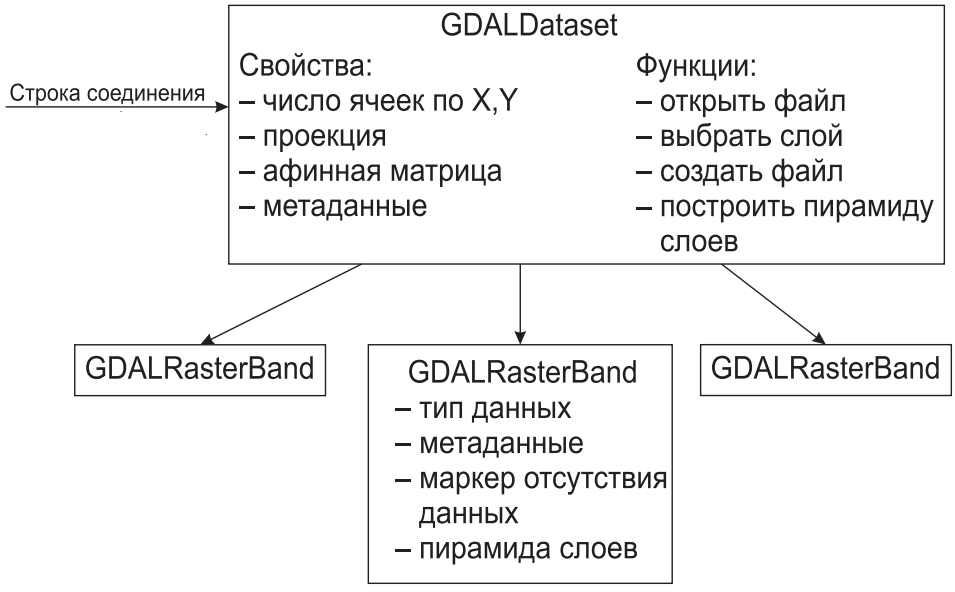

Рис. 1. Модель данных GDAL

данные о размере (ширина и высота), размере шага, географической привязке, координатной системе. Дополнительные данные хранятся в виде ключ–значение (метаданные). В качестве параметров открытия источника данных используется путь к файлу, строка соединения с базой данных или с веб-службой. В дальнейшем, используя слово «файл», подразумеваем, что аналогичные действия можно производить и для базы данных, и для веб-службы.

Система координат, хранящаяся в виде строк OpenGIS WKT (Well Known Text). Описывается с помощью класса OGRSpatialReference.

Растровый канал или слой описывается в GDAL с помощью класса GDALRasterBand. Растровый файл может содержать несколько слоев, однако некоторые форматы имеют ограничения. Например, формат AAIGrid (ASCII grid) может содержать только один слой, а JPEG – один или три (красный, зеленый, синий).

Таблица цветов состоит из нуля или нескольких записей, описываемых в виде структуры GDALColorEntry.

Обзорные изображения или пирамида слоев. Слой может содержать обзорные изображения, размер которых (в терминах строк и столбцов) будет отличаться от базового полноразмерного растра, однако географически они будут покрывать одну и туже область. Обзорные изображения применяются для быстрого отображения уменьшенных копий растра, вместо того, чтобы читать полноразмерное изображение с последующим масштабированием.

Создание драйвера GDAL для формата TOC4

Описание формата TOC, версия 4

Формат TOC (таблица объект-свойство) был разработан для системы ГИС ИНТЕГРО (разработка ФГУП ГНЦ РФ ВНИИгеосистем). Реализация поддержки данного формата позволила встроить библиотеку GDAL в среду ГИС ИНТЕГРО. На текущий момент с ее помощью производится конвертация данных. Ведутся работы по модификации механизма рисования карт в ГИС ИНТЕГРО, который позволит использовать форматы, поддерживаемые GDAL, напрямую без механизма импорта. В среде многофункционального геоинформационного сервера (МГС) драйвер позволил работать с файлами TOC без дополнительной конвертации, а так же интегрировать расчетные модули ГИС ИНТЕГРО (рис. 2).

Формат TOC4 представляет собой набор файлов: заголовок (имя_файла.pgrid), слои (имя_файла.имя_свойства.property), проекция в формате PROJ.4 (имя_файла_pgrid.pj4), данные о представлении слоя в ГИС ИНТЕГРО (имя_файла.имя_свойства._rg). Обязательными составляющими являются заголовок и набор данных.

Заголовок содержит:

- описание геометрии файла;
- количество слоев;
- дополнительную информацию о файле;
- блок описания слоев;
- имена слоев;
- заголовки слоев (названия слоев в программе);

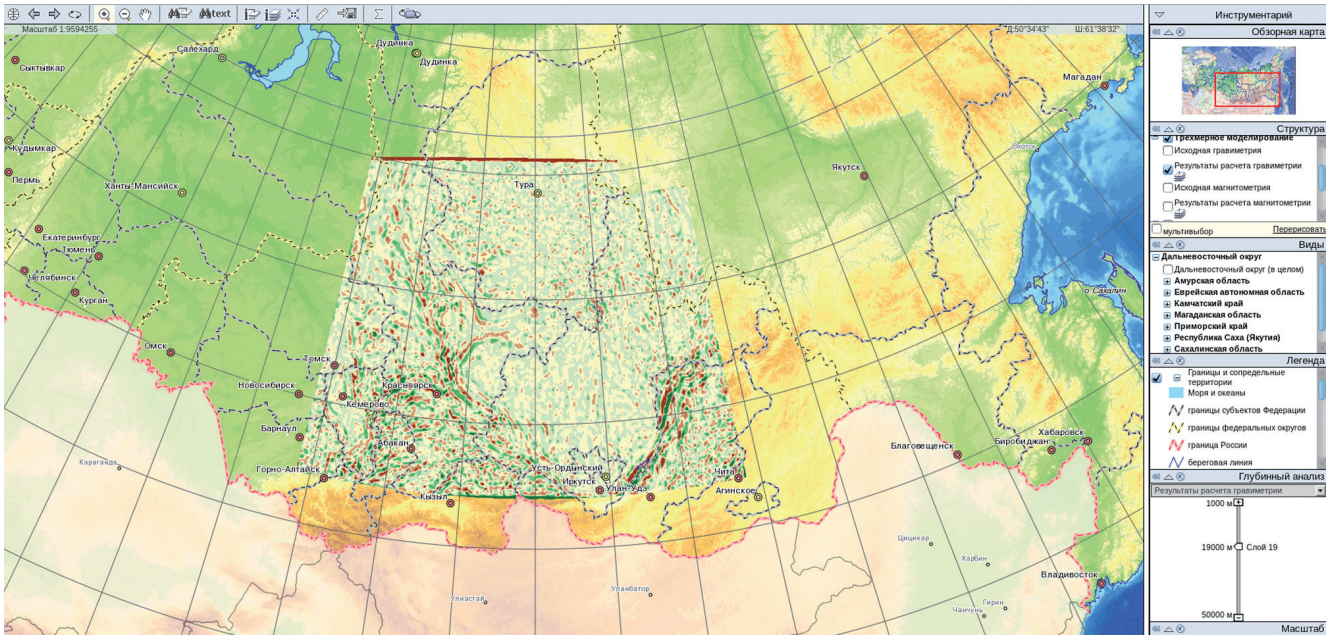

Рис. 2. Пример использования драйвера TOC4 в системе MGS

– типы слоев;
– типы данных в слоях (байт, целое, с плавающей точкой);
– статистические данные о слоях;
– гистограмму.

Процесс написания драйвера [2] можно условно разделить на два этапа. Первый – это реализация обязательных возможностей, которые включают в себя чтение данных из файла. Второй этап – реализация дополнительных возможностей, которые включают запись данных и работу с пирамидой слоев. Для работы в среде GDAL драйвер в обязательном порядке должен поддерживать только функцию чтения. Поддержка записи не обязательна.

Реализация функционала чтения

Чтение данных осуществляется в два этапа. На первом этапе считываются геометрические параметры источника данных и информация о слоях. На втором этапе уже происходит считывание данных.

Рассмотрим подробнее оба этих этапа.

Считывание информации об источнике данных библиотекой GDAL происходит методом последовательного поиска подходящего драйвера в коллекции зарегистрированных форматов путем последовательного перебора. Для этого вызывается функция Open() каждого драйвера. В качестве параметров передается строка соединения и около килобайта заголовка данных из файла. На основе этого заголовка принимается решение о продолжении разбора предложенного источника. В формате TOC заголовок совпадал с одним из XML-подобных форматов. Проблема была решена более жесткими условиями проверки.

Если формат опознан успешно, то далее производится чтение информации о файле. В модели данных библиотеки GDAL предполагается, что все слои в файле одного размера (геометрически). После считывания информации о геометрии создаются экземпляры данных слоя (GDALRasterband). Так же в функции Open() считываются размеры файла, информация о точке привязки, данные о проекции, метаинформация о файле и производятся другие необходимые настройки.

В случае драйвера TOC в конструктор слоя передается информация о пути к файлу данных слоя и дополнительная информация (имя, минимальное и максимальное значения данных, тип данных). В объявлении слоя так же необходимо указать размер считываемого блока. Для большинства форматов размером является одна строка данных. На данном этапе достаточно реализовать конструкторы GDALDataset, конструктор GDALRasterband, функцию Open() и функцию регистрации драйвера в системе (о ней речь пойдет ниже). Стоит отметить, что в GDAL точка начала координат – левый верхний угол.

Для проверки корректности работы драйвера можно вызывать утилиту gdalinfo, распространяемую вместе с библиотекой. Она выводит информацию об указанном файле. Если информация отображается корректно можно двигаться дальше.

Чтение данных слоя. Для чтения данных о слое используется функция IReadBlock(). В нее передаются смещения «в блоках» от начала координат и место, куда записывать полученный результат. Функция осуществляет чтение только одного блока. Данные считываются в формате, в котором записаны в файле. Преобразование данных в другие типы библиотека GDAL реализует сама.

Также необходимо реализовать функцию регистрации драйвера в системе GDALRegister_Имя_вашего_драйвера() и описать в ней возможности формата. Обязательно указывается название драйвера (короткое название, которое будет использоваться для вызова в библиотеке) и некоторая другая информация. Она описывается через механизм метаинформации. Необходимо указать параметры следующих переменных:

– GDAL_DMD_LONGNAME – длинное название драйвера;
– GDAL_DMD_EXTENSION – расширение;
– GDAL_DMD_CREATIONDATATYPES – если планируется записывать файл, то здесь перечисляются типы данных, в которых он может быть сохранен.

Так же здесь указываются функции, поддерживающие различные возможности драйвера (открытие соединения, запись данных).

Проверить корректность работы драйвера можно утилитой gdal_translate. Эта утилита производит преобразование данных из одного формата в другой.

Реализация функции записи в формат TOC4

Запись файла может осуществляться двумя способами:

– CreateCopy: результирующий источник данных создается из исходного, при этом изменение данных невозможно. Данный способ применяется в случае, когда невозможно определить позицию ячейки данных в файле, например, в ASCII-файлах или файлах со сжатием исходных данных. В качестве аргументов функция CreateCopy() принимает исходное соединение и параметры записи. Необходимо помнить, что исходные данные произвольного типа должны быть записаны в том же виде (если это поддерживается форматом).

– Create: источник данных создается в два шага. Сначала создается пустой каркас файла, со всеми параметрами (шагом, проекцией и прочим), заполненный отсутствующими значениями (nodata). Запись данных осуществляется позже. Каркас файла создается функцией Create(). Запись блока данных осуществляется функцией IWriteBlock(). В данной функции необходимо учитывать, что исходные данные произвольного типа необходимо преобразовывать к типу создаваемого источника данных.

Расширенные возможности драйвера

Для ускорения доступа к данным используется такой механизм – пирамида слоев (обзорные изображения). В основном он применяется при отрисовке слоя данных. Он предоставляет собой дополнительные слои, в которых данные хранятся в разряженном состоянии. При использовании GDAL можно запросить прямоугольный участок слоя и записать его в массив с кратно уменьшенными размерами. В случае, когда пирамида слоев не задана, разряжение производится средствами библиотеки. В случае наличия пирамиды автоматически определяется самый ближний слой и данные берутся из него.

Реализация работы с обзорными изображениями

Для работы с обзорными изображениями необходимо добавить в GDALDataset функции создания и удаления пирамиды (IbuildOverviews() и CleanOverviews()) и в функцию открытия файла Open() добавить инициализацию текущей пирамиды источника данных oOvManager.Initialize().

Для слоя необходимо реализовать функции выбора слоя пирамиды GetOverview() и подсчет построенных слоев GetOverviewCount(). При создании слоя необходимо открывать файл с пирамидой.

Формат TOC не предусматривает хранение пирамиды слоев. Поэтому при его разработке был использован вариант, когда пирамида хранится в виде файла в формате GeoTiff по аналогии с файлами, содержащими информацию о данных слоя (имя_файла.имя_свойства.ovr).

Модель данных OGR

Модель данных (рис. 3) OGR описывает программное представление структуры векторных данных. Драйвер для работы с векторным форматом состоит из 3-х частей: описания драйвера, описания источника данных и описания слоя в источнике.

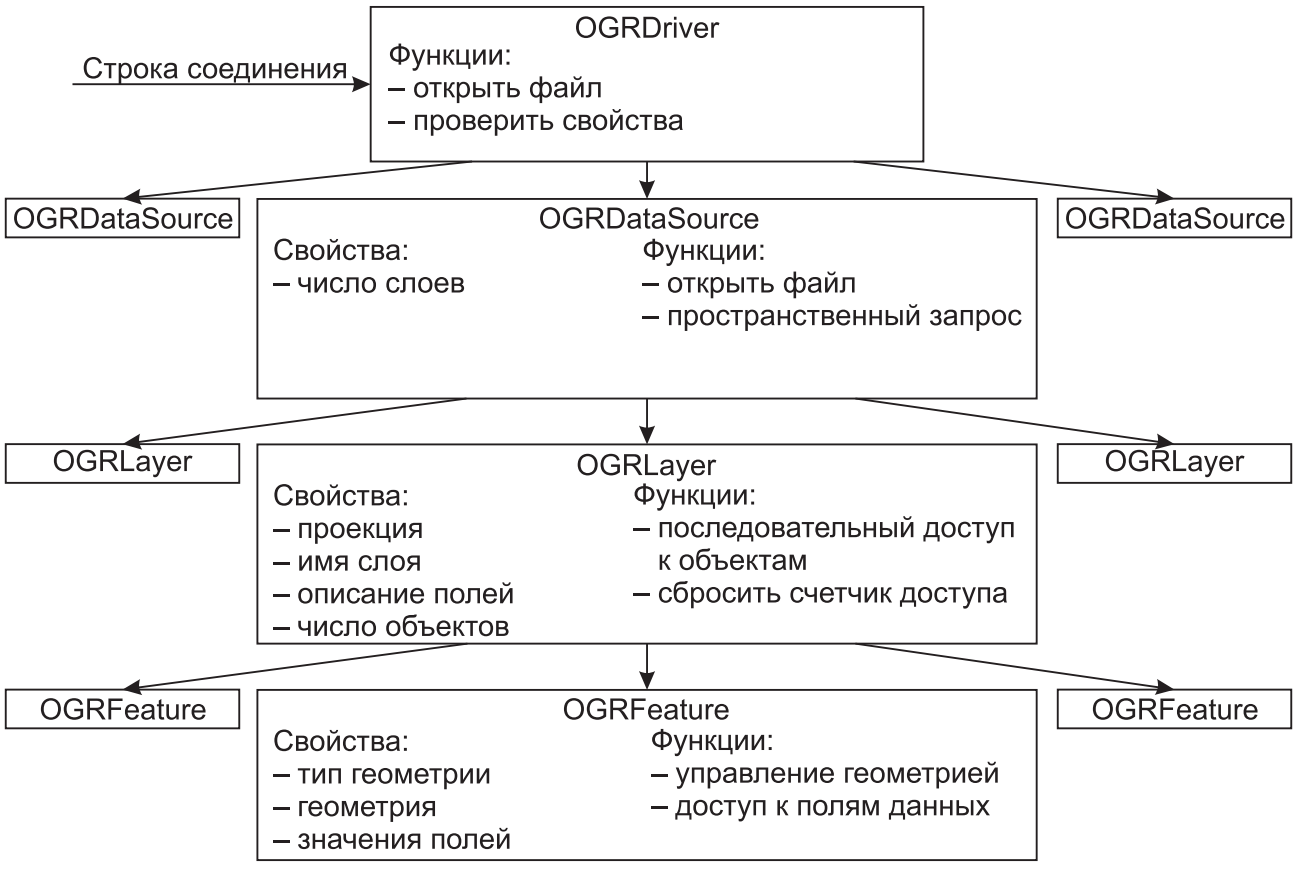

Рис. 3. Модель данных OGR

Каждая из них должна содержать описание возможностей по работе с данным блоком, которая описывается в функции TestCapability(). При работе используются следующие понятия:

Driver (драйвер) описывается классом в формате НазваниеФорматаDriver. Содержит механизмы регистрации драйвера в библиотеке. Обязательно должен поддерживать функцию Open(), в которой происходит идентификация и открытие источника данных. Дополнительно возможно реализовать создание источника данных или его удаление. При открытии векторного файла, так же как и при открытии растрового, библиотека последовательно обходит все форматы и проверяет их на соответствие открываемому источнику данных.

Драйвер может поддерживать следующие возможности:

- создание источника данных;
- удаление источника данных;
- чтение источника данных.

По умолчанию считается, что любой источник данных можно читать.

DataSource (источник данных) описывается классом в формате НазваниеФорматаDataSource. Содержит инструменты по работе с источником данных, обеспечивает доступ к слоям.

В функции Open() открывается источник данных, определяются его параметры (число объектов, проекция, информация о полях данных и другие), открываются слои. Получение доступа к конкретному слою осуществляется через функцию GetLayer().

Слой поддерживает следующие возможности:

- чтение;
- создание;
- удаление.

По умолчанию считается, что любой слой можно читать.

Layer (слой) описывается классом в формате НазваниеФорматаLayer. Содержит инструменты по работе со слоем, функции доступа к объектам и полям атрибутов.

При открытии слоя считывается информация об охватывающей рамке, если она содержится в файле, и общая информация об объекте.

Работа со слоем поддерживает следующие возможности:

- непоследовательный доступ к объектам;
- последовательная запись объектов;
- непоследовательная запись объектов;
- быстрая фильтрация объектов по поисковому запросу;
- быстрый подсчет количества объектов;
- быстрое получение охватывающей рамки;
- создание поля в таблице свойств;
- транзакции;
- удаление объекта.

Стандартный способ доступа к объектам – это последовательный перебор. Он осуществляется функцией GetNextFeature(). Внутри нее создается объект OGRFeature, который содержит геометрию и атрибуты. Функция возвращает объект, а не указатель на него, что влечет необходимость его удаления после использования.

Для сброса порядка обхода используется функция ResetReading().

Для корректной работы также должна поддерживаться функция GetExtent(), позволяющая получить охватывающую рамку источника данных.

Создание драйвера OGR для формата SXF (ГИС Панорама)

Описание формата SXF

SXF (Storage and eXchange Format) [3] – открытый формат цифровой информации о местности. Предназначен для применения в геоинформационных системах для хранения цифровой информации о местности, обмена данными между различными системами, создания цифровых и электронных карт и решения прикладных задач. Он является основным инструментом создания электронных карт в формате SXF – ГИС «Карта 2008» и ГИС «Карта 2011» (ЗАО КБ «Панорама»). В формате SXF осуществляется создание и хранение цифровой картографической продукции Росреестра, в том числе цифровых навигационных карт ФЦП «ГЛОНАСС». В настоящее время обсуждается возможность использования формата SXF для хранения и обмена цифровыми топографическими и навигационными картами и планами городов, создаваемыми за счет средств федерального бюджета. Однако работа с форматом в сторонних ГИС, не принадлежащих разработчику формата, реализована слабо. Бесплатных библиотек по работе с форматом или конвертеров в общепринятые форматы найти не удалось. Поэтому было принято решение разработать драйвер поддержки формата для библиотеки OGR.

Файл формата SXF может содержать в себе данные нескольких слоев с различными типами геометрии (точки, линии, полигоны). По спецификации данные лежат последовательно, без разделения по слоям или типам, что затрудняет работу с ними. Отношение данных к одному слою можно определить по коду классификатора, поэтому при работе драйвера на этапе открытия проводится поиск уникальных идентификаторов классификатора с разделением по

типу геометрии. На основе этого формируются слои. Также в момент открытия слоя строится для всех объектов индексная таблица, на основании которой производится поиск конкретного объекта при его запросе.

Файл SXF[4] состоит из:

- заголовка (паспорта), который содержит информацию о структуре файла, проекции данных, охватывающей рамке, системных переменных;
- дескриптора данных, содержащего информацию о способе записи данных (число записей, способ записи координат, классификаторе данных и другое);
- блоков данных, содержащих таблицу заголовка записи и таблицу координат;
- блока семантики объекта – запись таблицы данных со свойствами объекта. Значения семантики могут быть ссылками на значения полей классификатора.

Совместно с SXF используется файл классификатора, который содержит дополнительную информацию о данных (описание слоев, значение атрибутов, визуальное представление объектов и другое). Файл классификатора имеет расширение .rsc и может использоваться для большого числа однотипных файлов SXF.

Структура классификатора:

- заголовок данных, содержащий служебную информацию о структуре файла;
- таблицы, содержащие информацию о структуре классификаторов и данные;
- структуры, описывающие графическое оформление (работа с этими структурами не реализовывалась в рамках драйвера OGR).

Хотя в стандарте SXF принято, что одному файлу классификатора может соответствовать несколько карт, для однозначной идентификации было принято решение, что рядом с файлом данных (.sxf) находится одноименный файл классификатора (.rsc).

Пример использования драйвера SXF в системе МГС приведен на рис. 4.

Сравнение скорости доступа к данным в различных форматах через библиотеку GDAL/OGR

Эксперимент производится на компьютере: CPU AMD Phenom II x6 1075T, 4Gb RAM.

Сравнение скорости чтения растровых форматов

В эксперименте проверялась скорость чтения данных. В тесте участвовали форматы: GeoTiff, AIG (ArcInfo binary grid), AAIGrid (ArcInfo ASCII grid), TOC4 (ГИС ИНТЕГРО). Размер тестируемого файла 2777 на 2682 точек. Данные записаны в целочисленном формате.

Для удаления погрешностей измерения времени эксперимент проводился 100 раз. За шаг эксперимента производились следующие действия:

- открытие файла данных;
- считывание данных в массив, расположенный в памяти (память для массива выделялась за пределами цикла эксперимента);
- закрытие файла.

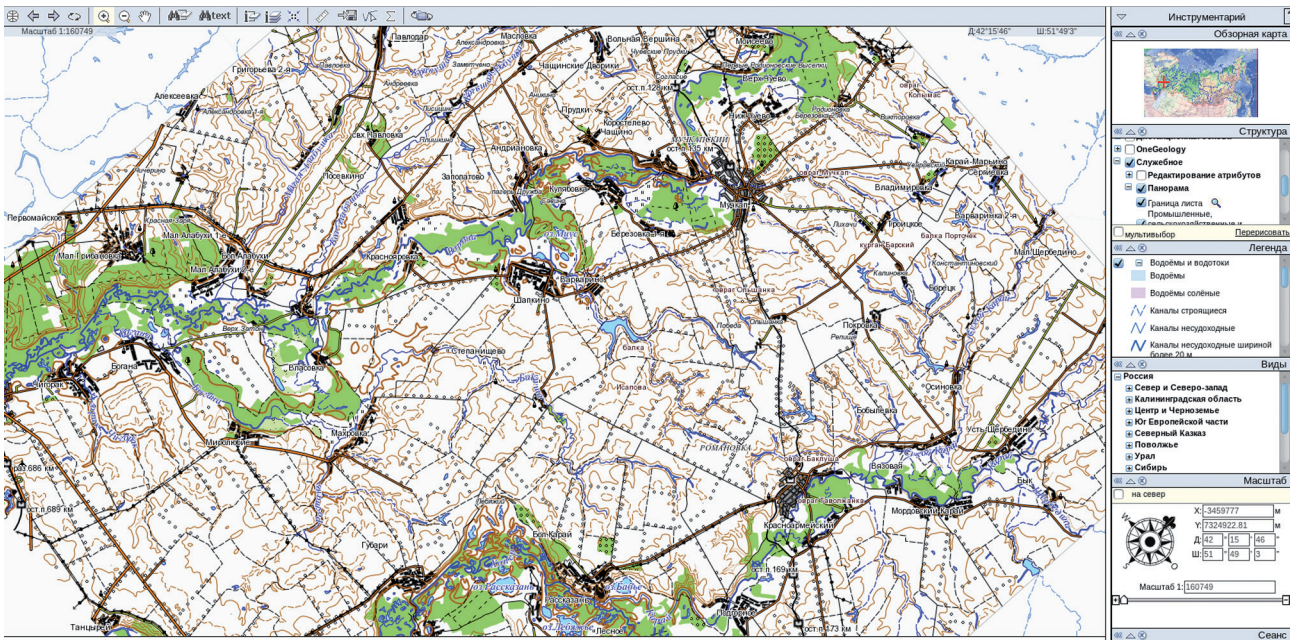

Рис. 4. Пример использования драйвера SXF в системе МГС

Проводилось четыре теста:

1. Чтение всего объема данных в целочисленном формате.
2. Чтение файла с преобразованием типа к данным с плавающей точкой.
3. Чтение данных по строкам в целочисленном формате.
4. Чтение данных по столбцам в целочисленном формате.

Результат эксперимента представлен в табл. 1 и на рис. 5.

Замедление чтения по столбцам обосновано тем, что в драйверах эта операция производится построчно. В этом случае на каждый шаг по Y читается строка данных. Замедление скорости чтения для формата TOC4 связано с тем, что открытие файла свойств происходит на каждое обращение к блоку данным. Это обусловлено требованием совместимости с ГИС ИНТЕГРО, где по требованию архитектуры файл должен быть всегда закрыт.

Резкое замедление скорости чтения в формате AAIGrid связано с тем, что доступ к ASCII-данным медленнее, чем к бинарным.

Сравнение скорости для векторных форматов

Тестирование производилось на трех форматах: ESRI Shapefile, MapInfo, SXF.

Файл ESRI Shapefile содержал один слой, MapInfo и SXF – полный набор слоев карты на Воронежскую область (81 слой, включающий в себя точечные, линейные, векторные слои и слой подписей).

Тестирование производилось для линейного слоя с изолиниями высот и полигонального слоя с кварталами городов. Слой высот содержал 206 объектов, кварталов – 2558.

Так же как и для растровых форматов, эксперимент проводился 100 раз.

Результат эксперимента представлен в табл. 2 и на рис. 6.

Замедление скорости чтения формата SXF связано с отсутствием внутреннего индекса для доступа к объектам и внутреннего деления на слои. В связи с чем при загрузке файла необходимо искать уникальные индексы классификатора для объединения объектов в слои и строить индексную таблицу объектов для каждого слоя.

Таблица 1

Результат эксперимента скорости доступа к растровым форматам

Номер теста	GeoTiff, сек	AIG, сек	AAIGrid, сек	TOC4, сек
1	4,0	4,4	62	12,8
2	4,5	5,2	60	13,7
3	4	4,8	60,7	13
4	126	42,7	176	133

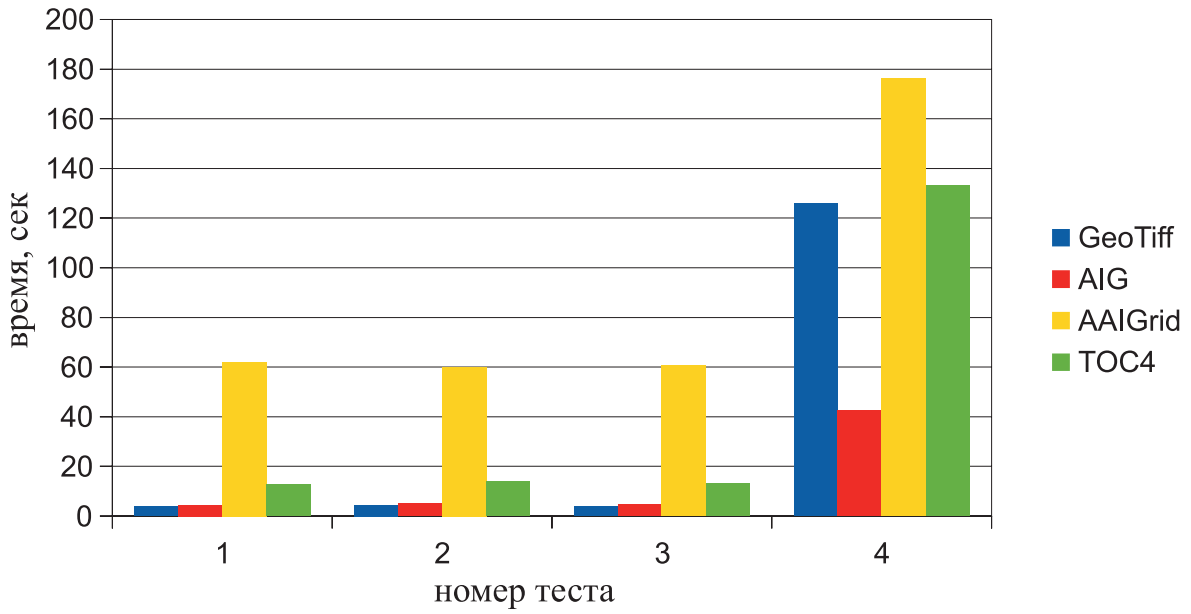

Рис. 5. Сравнение скорости доступа к форматам в библиотеке GDAL

Таблица 2

Результат эксперимента скорости доступа к векторным форматам

	SHP (ESRI Shapefile), сек	MapInfo file, сек	SXF (ГИС Панорама), сек
1. Линейный слой	0,4	1	11,4
2. Полигональный слой	4,2	2	13,3

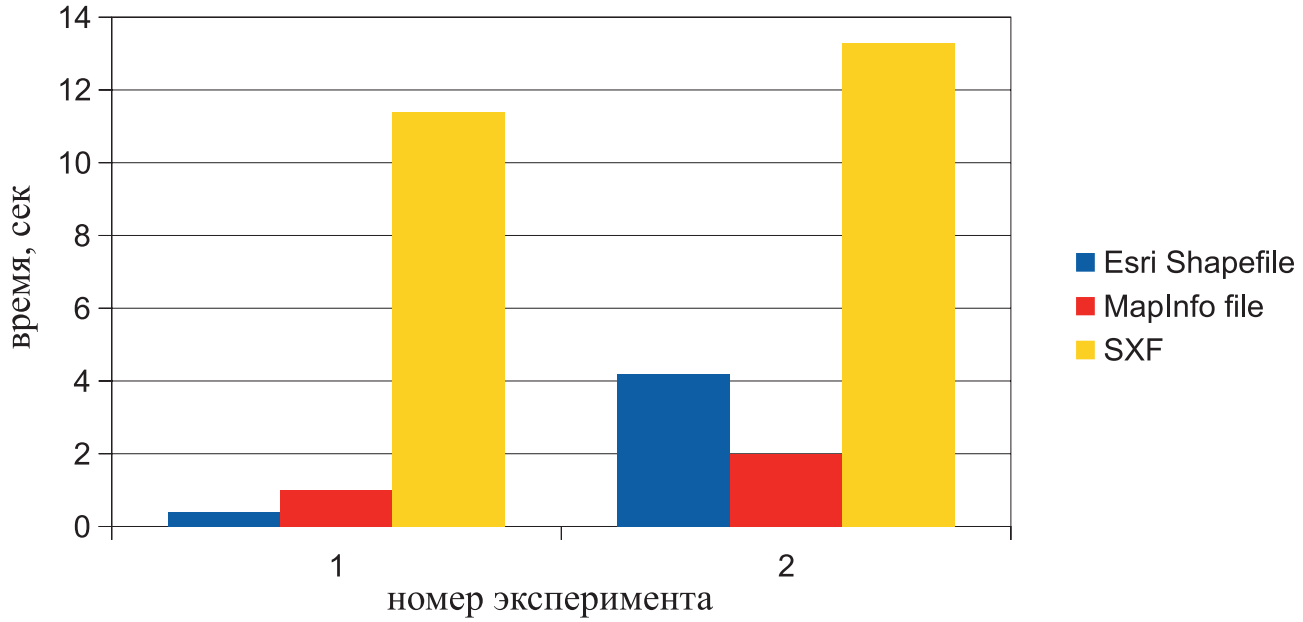

Рис. 6. Сравнение скорости доступа к форматам в библиотеке OGR

Заключение

Применение библиотеки GDAL/OGR позволяет снизить время и стоимость разработки средств доступа к различным форматам данных ГИС. Реализация собственных драйверов форматов дает возможность использовать механизмы универсального доступа к данным не только для одного продукта, но и для всех остальных, которые используют данную библиотеку.

Вышеперечисленные драйверы созданы для использования в программных продуктах, разрабатываемых во ВНИИгеосистем. Библиотека GDAL/OGR в ГИС ИНТЕГРО используется для обеспечения импорта и экспорта широкого спектра форматов данных, а многофункциональный геоинформационный сервер – в качестве основного средства доступа к геоданным.

Дальнейшее развитие работ предполагает улучшение скорости доступа к данным формата TOC и SXF. В случае принятия формата SXF как стандарта для хранения и обмена цифровыми топографическими и навигационными картами будет реализован механизм записи в данный формат.

Ключевые слова: программирование, форматы данных, драйвер доступа, GDAL/OGR

ЛИТЕРАТУРА

1. Модель данных GDAL // GDAL – Geospatial Data Abstraction Library. – URL: http://www.gdal.org/gdal_datamodel_ru.html (дата обращения 02.08.2012).
2. GDAL Driver Implementation Tutorial // GDAL – Geospatial Data Abstraction Library. – URL: http://gdal.org/gdal_drivertut.html (дата обращения 02.08.2012).
3. OGR Driver Implementation Tutorial // OGR Simple Feature Library. – URL: http://gdal.org/ogr/index.html (дата обращения 02.08.2012).
4. Векторный формат «SXF» структура данных в двоичном виде. Редакция 4.0 // КБ Панорама Геоинформационные технологии. – URL: http://gistoolkit.ru/download/doc/sxf4bin.pdf (дата обращения 02.08.2012).