

Преобразования координат в геоинформационных системах

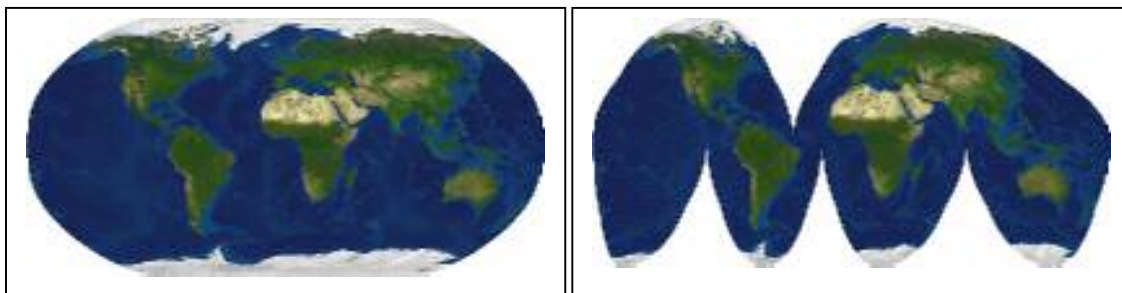
Лебедев К.Г.

В докладе рассматриваются проблемы преобразования координат и связанные с ними преобразования растровых изображений картографических документов. Для решения указанных задач автором разработана объектно-ориентированная библиотека функций, апробированная в одной из коммерческих геоинформационных систем.

Основой преобразования координат является класс `CoordinateSystem`. Сам этот класс реализует координатную систему широта/долгота. Класс имеет три виртуальные функции предназначенные для преобразования координат. Функция `Prepare` обеспечивает предварительные вычисления для последующего преобразования координат при заданных параметрах проекции. Функция `ConvertForward` осуществляет пересчёт координат точек, заданных широтой и долготой, в значения проекции. Соответственно функция `ConvertInverse` осуществляет обратное преобразование. Если нам нужно осуществить пересчёт координат из проекции Меркатора в проекцию Робинсона нам потребуется экземпляр класс `CoordinateSystemMercator` (с заданными параметрами проекции) и экземпляр `CoordinateSystemRobinson` (так же с параметрами). Предварительно для каждого из классов проекция вызывается функция `Prepare`, которая выполняет подготовительную работу перед началом преобразования (рассчитывает коэффициенты, которые зависят только от параметров проекции и не зависят от точек подлежащих проектированию). После этого точки заданные в проекции Меркатора функцией `CoordinateSystemMercator::ConvertInverse` переводятся в точки, заданные широтой/долготой, и затем функцией `CoordinateSystemRobinson::ConvertForward` переводятся в проекцию Робинсона. Но этот путь не учитывает тот факт, что любая проекция включает в себя в качестве параметра геоид, на котором она задана, и что при несовпадении геоидов для проекция необходимо переводить точки заданные на одном геоиде в точки, заданные на другом. Поэтому для практической работы создан класс `CoordinateConverter`, которые получает указатели на экземпляры классов двух проекций (`Source` и `Target`), и с помощью единственной функции `Convert` производит перевод точек из первой проекции во вторую, при необходимости осуществляя пересчёт точек из геоида одной проекции на геоид другой (при пересчёте используются формулы Молоденского).

Теперь предположим, что нам необходимо спроектировать растровое изображение из одной системы координат в другую. При попиксельном проектировании могут возникнуть различные варианты размещения пикселей исходного изображения на пикселах целевого. Так, например, несколько пикселей исходного изображения могут проектироваться в один пиксел целевого, а в некоторые пиксела целевого может не проектироваться не один пиксел исходного. Если первая ситуация имеет хорошее решение, цветом пиксела целевого изображения будет средний цвет проектировавшихся в него пикселей исходного изображения, то в случае отсутствия пикселей исходного изображения проектировавшихся в пиксел целевого хорошее решение предложить не просто (интерполирование цвета от ближайших сложно, и непонятно как интерполировать на краях изображения). Поэтому при проектировании растровых изображений используется оригинальный подход. Применяется обратное проектирование, т.е. пиксела целевого изображения проектируются в пиксела исходного, и после этого цвет пиксела исходного изображения устанавливается путём билинейной интерполяции в исходном изображении, если же пиксел целевого изображения не попадает в исходное изображение, то его цвет так и остаётся не заданным.

Для того чтобы определить размер целевого изображения предварительно всё же пиксели исходного изображения проектируются в систему координат целевого, и высчитывается размер охватывающего их прямоугольника, который и будет являться размером целевого изображения.



На рисунках изображён результат проектирования растрового изображения из проекции Robinson в проекцию Goode's Homolosine.

Подход к проектированию растровых изображений.

При работе с гео-привязанными данными часто возникает потребность перевода их из одной системы координат в другую. Например исходные данные бывают доступны в различных координатных системах, а работа с ними должна осуществляться в одной. Если перевод векторных данных не вызывает затруднений при наличии формул для пересчёта из системы в системы, то перепроектирование растровой информации (космическая и аэрофотосъёмка, сканированные бумажные карты) задача гораздо более сложна. В работе предлагается базовая архитектура классов для осуществления перепроектирования данных между координатными системами. Каждая координатная система реализуется отдельным классом и содержит методы для пересчёта из значений широты/долготы в значения проекции и обратно. Кроме того, создаётся класс для пересчёта координат в значения широты/долготы заданных на одном геоиде в значения координат на другом геоиде, так же в значения широты/долготы. Теперь для того, чтобы перепроектировать значение из одной координатной системы в другую, достаточно создать экземпляры объектов реализующих эти координатные системы, перепроектировать с помощью объекта реализующего исходную систему значения заданные в исходной системе в значения широты/долготы, при несовпадении геоидов в системах осуществить пересчёт координат между ними, и с помощью объекта второй системы пересчитать значения широты/долготы в значения целевой координатной системы. Для перепроектирования же растровых изображений предлагается оригинальный подход. Не попиксельное проектирование исходного изображения в целевое (согласно описанной выше схеме), а проектирование пикселей целевого изображения в исходное (назовём это обратным проектированием). Т.е. нахождения для каждого пикселя целевого изображения его прообраза на исходном, и выставлении цвета этого пикселя в соответствии с его прообразом используя билинейную интерполяцию по ближайшим пикселям. Подобный подход позволяет получать результаты которые невозможно получить и в некоторых коммерческих продуктах.