

# Обновление карт текущей обстановки на основе регулярно поступающих космических изображений земной поверхности.

Непомнящий П.В.  
ЦОС и ВТ МФТИ  
Москва, Россия

## Аннотация

Разработана программная библиотека хранения и регулярно обновления карт заданного участка земной поверхности на основе регулярно поступающих аэрокосмических изображений со спутника. Карты могут иметь несколько слоев, хранящих разные виды данных. Система позволяет переводить поступающие изображения в требуемую географическую проекцию с минимальной потерей точности, осуществлять композицию данных с нескольких слоев в другой слой, производить извлечение и замену участков карты. Ввиду того, что карты высокого разрешения занимают значительный объем, для оптимизации использования машинных ресурсов они хранятся разрезанными на множество небольших изображений, что позволяет загружать в память только тот участок изображения, с которым ведется работа в данный момент.

**Ключевые слова:** аэрокосмические изображения, географические проекции, карты, трансформация изображений, обновление данных, мониторинг.

## 1. ВВЕДЕНИЕ

После приема и обработки изображений земной поверхности со спутника, возникает задача их хранения и обновления. Сеансы со спутника поступают регулярно и нужно поддерживать данные в соответствии с ними. Часто нужно производить мониторинг какого-либо участка земной поверхности (например, области города, для предсказания погоды). Сеансы со спутника, как правило, охватывают требуемую область не целиком, а частично, поэтому накопление информации по одной области требует данных с нескольких сеансов. Также принимаемые космические изображения сильно искажены, за счет сферичности земли, особенностей съемочной аппаратуры и ориентации спутника [3]. Поэтому, прежде чем совмещать несколько изображений требуется перевести их в единую систему координат – как правило, в этой роли выступает одна из распространенных географических проекций [2].

После поступления нового изображения и перевода его в географическую проекцию требуется обновить уже существующую карту данного участка земной поверхности, наложив на неё это изображение. Тут следует учитывать не только то, что новое изображение может покрывать заданную область не полностью, но и то, что часть земной поверхности на изображении может быть скрыта облаками, помехами. В этом случае целесообразно производить наложение по маске, таким образом, чтобы то место, на котором на новом изображении находятся облака или помехи, не обновлялось, и на нем остались старые данные.

Для дальнейшей обработки интерес представляют не только оригинальные спектральные каналы, получаемые со спутника, но и результаты их тематической обработки – температурная карта поверхности, обнаружение облаков, водоемов, индекс вегетации - то есть карта в общем случае может содержать много слоев с разными типами данных, причем часто возникают ситуации, когда с привлечением данных нескольких существующих слоев формируется новый слой (например, наложение на новый сеанс маски облаков). В общем случае это задача композиции нескольких слоев в один по заданному закону.

## 2. СТРУКТУРА БИБЛИОТЕКИ

Для решения задач, возникающих при отслеживании состояния участка земной поверхности, была разработана библиотека классов C++, предоставляющая средства для хранения и обновления карт земной поверхности.

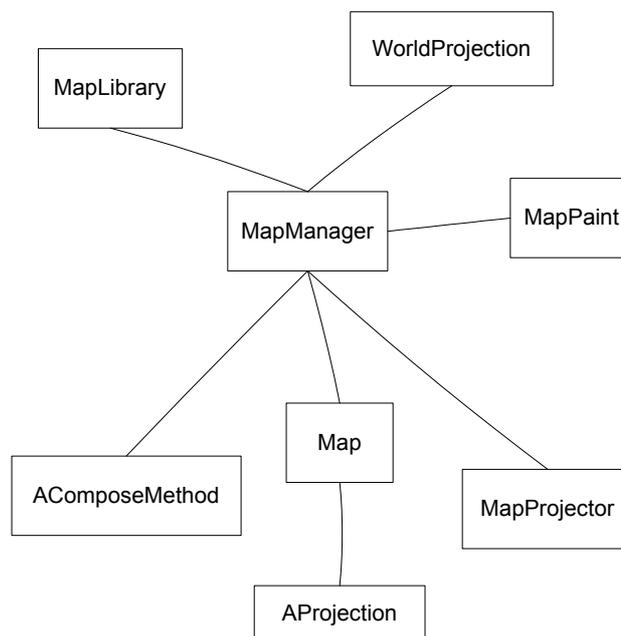


Рисунок 1. Общая структура библиотеки

На самом верхнем уровне структуры хранения карт находится класс библиотеки карт (MapLibrary) (рис. 1), который отвечает за хранение всех существующих в данной системе карт. Каждая карта задается уникальным идентификатором,

названием, географической проекцией, областью земной поверхности которую она описывает и масштабом.

Класс MapLibrary предоставляет доступ к картам, их создание и удаление, обеспечивает их хранение на носителе информации (рис. 2).

Для работы с отдельной карты существует класс MapManager, ссылку на объект которого выдает по запросу с указанием уникального идентификатора класс MapLibrary. Этот класс отвечает за хранение и модификацию карт.

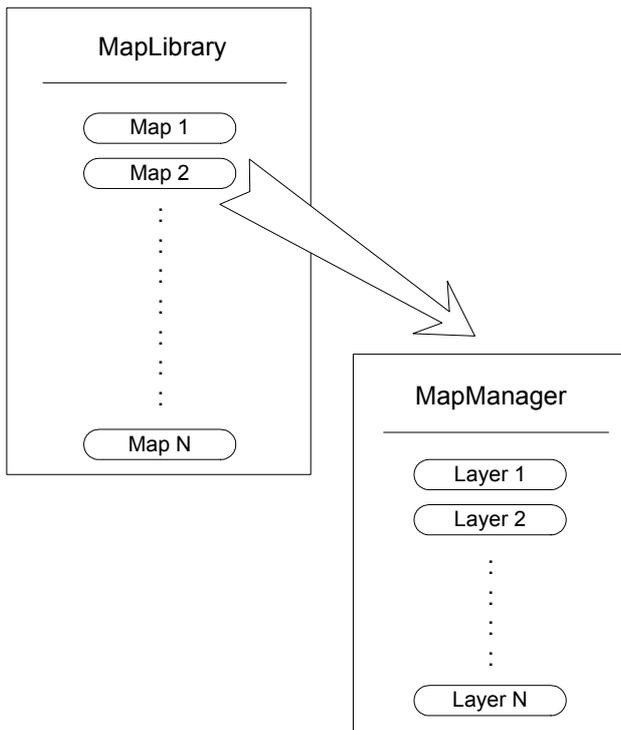


Рисунок 2. Состав карт и библиотеки карт

Для представления самой карты используется класс Map. Он описывает географическую проекцию, область, покрываемую картой и масштаб. Также этот класс предоставляет функции перевода географических координат в координаты проекции и обратно.

Проекция в библиотеку можно добавлять по мере необходимости. Все они должны быть унаследованы от класса AProjection, который является абстрактным и предоставляет интерфейс для реализации производными классами.

Класс AComposeMethod определяет интерфейс для объектов-функций, описывающих композицию слоев. Композиция слоев – это получение по заданным правилам нового слоя карты из нескольких существующих слоев.

Для проецирования изображения мировой карты в проекции Plate Carree в другие проекции предназначен класс WorldProjection.

### 3. УПРАВЛЕНИЕ КАРТАМИ

Для работы с картами библиотека карт предоставляет специальный объект (MapManager) (рис. 1), который обеспечивает такие функции как получение данных со слоев карты, проецирование новых сеансов, композицию слоев карты, добавление и удаление слоев, извлечение и запись фрагментов слоя карты.

Модифицирующие операции, которые выполняет библиотека (проецирование, композиция), выполняются не над основными данными, а над временно создающимися слоями, что позволяет при необходимости отменять произведенную операции с сохранением старых данных. Если результат операции удовлетворительный, временный слой замещает основной.



Рисунок 3. Схема обновления данных при поступлении нового сеанса со спутника

#### 3.1 Проецирование поступающих сеансов

При поступлении нового сеанса со спутника, после тематических обработок, следующим этапом является его перевод в ту географическую проекцию, в которой представлена данная карта (рис.3).

Поскольку цифровые изображения дискретны, всякое преобразование приводит к накоплению ошибки, за счет того, что пиксели исходного и результирующего изображения практически никогда не попадают друг на друга и приходит-

ся использовать аппроксимацию по близлежащим пикселям [4,5]. Поэтому преобразование желательно выполнить за один этап, с использованием методов аппроксимации, дающих минимальную погрешность. Для этой цели был использован разработанный ранее алгоритм [1] перевода аэрокосмических изображений в географическую проекцию. Этот алгоритм производит выбор способа аппроксимации на основе локального коэффициента сжатия, что минимизирует накопление ошибки, а также позволяет производить перевод в географическую проекцию сразу нескольких слоев одновременно, что существенно уменьшает затраты времени. На данный момент реализованы следующие проекции:

- Mercator
- Transverse Mercator
- Universal Polar Stereographic
- Lambert Equal Area Azimuthal
- Equidistant Cylindrical
- Plate Carree

Проекцией является преобразование координат вида:

$$\begin{aligned} x' &= \varphi(x, y) \\ y' &= \psi(x, y) \end{aligned} \quad (1)$$

Это преобразование переводит географические координаты (ширину и долготу) в координаты проекции, причем угол поворота и масштаб преобразования не постоянен и зависит от координат [2]. Для проекции Меркатора преобразование имеет вид:

$$\begin{aligned} x &= ax' \cos(P) \\ y &= a \ln \left( \tan \left( \frac{\pi}{4} + \frac{y'}{2} \right) \right) \cos(P) \end{aligned} \quad (2)$$

Где  $P$  – стандартная параллель, параметр проекции,  $a$  – большая полуось эллипсоида.

Новый сеанс проецируется всегда в отдельный временный слой, чтобы не нарушать целостность уже существующих данных.

### 3.2 Обновление данных, композиция

После того, как изображение спроецировано на отдельный слой, нужно произвести его композицию с уже существующими данными. Это может быть как просто наложение нового сеанса на старое изображение, так и более сложная композиция с учетом облачного покрова и других факторов (рис. 3).

Библиотека предоставляет возможность пользователю создавать свои собственные правила композиции, оформляя их в виде отдельных объектов-функций с унифицированным интерфейсом. Объект-функция должен предоставлять по запросу информацию о том, какие слои требуются для осуществления композиции, какие желательны, но не обязательны, а также функцию, которая производит композицию на основе предоставленного массива значений пикселей заданных слоев и возвращает результат композиции (рис. 4).

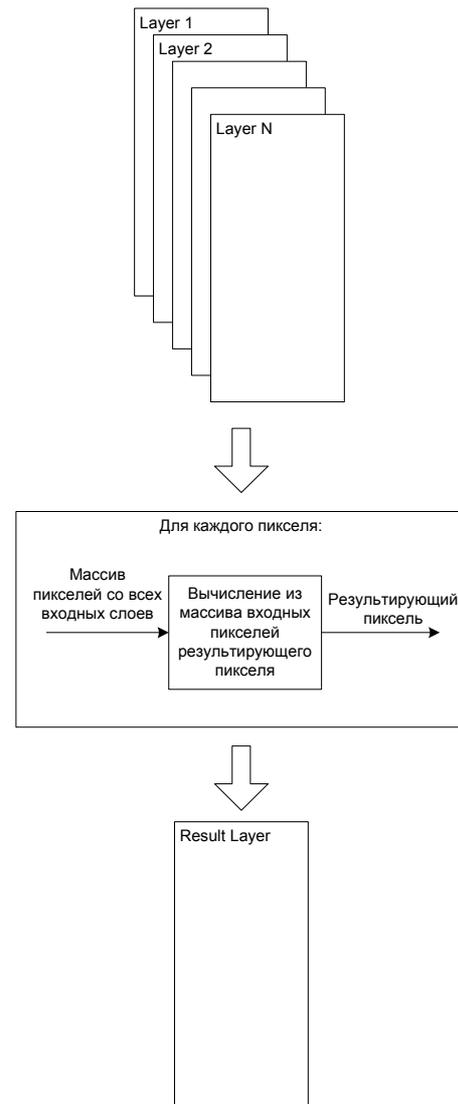


Рисунок 4. Процесс композиции нескольких слоев в один

Абстрактный класс, который описывает интерфейс объекта-функции выглядит следующим образом:

```
class AComposeMethod
{
public:
virtual ~AComposeMethod () {}
// Оператор (), который осуществляет композицию.
virtual const unsigned int*
operator()(const unsigned int* values) const=0;
//Получение массива слоев, требуемых для композиции
virtual vector<pair<int,bool> > getInputLayers() const=0;
};
```

Реализация функции **operator()** для композиции двух слоев, старого и нового, с учетом маски облаков (старый слой за-

мещается новым, если на этом месте на новом слое нет облаков) выглядит следующим образом:

```
const unsigned int* operator()(const unsigned int* values) const
{ //предполагается что values – это массив пикселей, где
  // values[0] – интенсивность старого слоя
  // values[1] – интенсивность нового слоя
  // values[2] – значение маски облаков, если не 0 – то есть
  // облака
    if(values[2])
        return values[0];
    else
        return values[1];
}
```

Результат композиции также помещается во временный слой, для оценки и принятия решений о его утверждении или отмены. После этого, если решение положительно, новый слой принимается и замещает старый слой.

### 3.3 Извлечение и запись фрагментов, операции со слоями

Объект для работы с картой также позволяет, для случаев, когда возможностей, предоставленных композицией, недостаточно, извлекать и вставлять фрагменты изображений слоев напрямую. Это может применяться, например, для нанесения на карту какой-либо дополнительной информации, обозначений, названий населенных пунктов.

Также объект для работы с картой позволяет добавлять и удалять существующие слои.

## 4. ХРАНЕНИЕ КАРТЫ

### 4.1 Разбиение на подкарты

Кроме выполнения операций с картой, объект MapManager обеспечивает хранение карты на носителе информации. В силу того, что разрешение спутника, с которого получают изображения, порядка 1км/пиксель, а в дальнейшем планируется принимать сеансы со спутников чья разрешающая способность еще выше, изображения земной поверхности имеют очень большой объем, порядка десятков и сотен мегабайт (типовой сеанс со спутника серии NOAA имеет разрешение 2048 пикселей в ширину и порядка 4000-6000 пикселей в длину). В то же время, ясно, что для обработки отдельного участка изображения не требуются данные для всего изображения. Но в случае, когда весь слой карты хранится в одном изображении, для изменения отдельного участка приходится считывать все изображение в память. Это приводит к неоправданному расходу памяти, что может замедлить работу системы. Для того, чтобы оптимизировать использование памяти, исходная карта разбивается на множество подкарт, и изображения хранятся кусочно (рис.5), для каждой подкарты – отдельное. В случае внесения изменений в карту, в память подгружаются только те изображения, которые непосредственно изменяются, или данные о которых нужны для проведения изменений. Также такой подход позволяет не создавать пустые изображения, информация для которых еще не получена.

Это разбиение карт на подкарты прозрачно для пользователя, то есть, оно происходит незаметно для него. Сетка координат для разбиения используется одна и та же при одина-

ковом масштабе и географической проекции, что дает дополнительную возможность совмещать различные карты, например с разных станций приема сеансов, если они выполнены в одной проекции и масштабе.

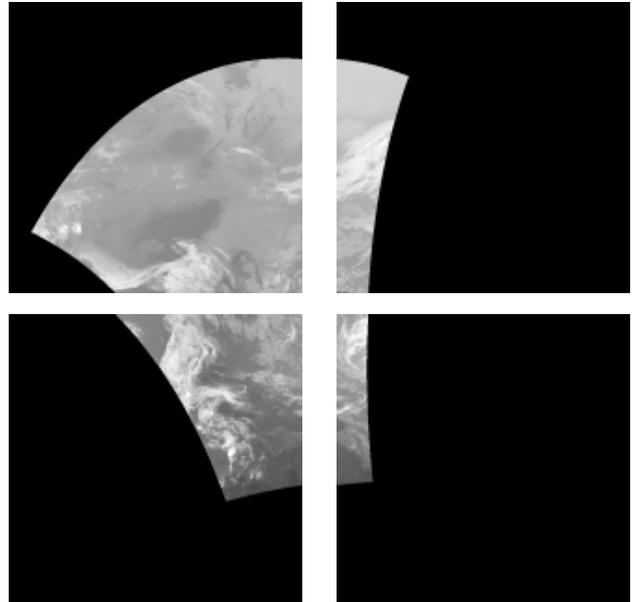


Рисунок 5. Разбиение карты на подкарты

Для отображения карт на экран в разных масштабах изображения хранятся иерархически, с коэффициентами сжатия от 1 до 10, что позволяет при отображении больших областей карты не загружать в память и не перерастрировать большой объем изображений.

При композиции, если на требуемой подкарте нет данных, то пустое изображение не создается, а подставляется нулевое значение, что также экономит использование памяти.

### 4.2 Слои карты

Карты содержат в себе много слоев, которые хранят разные типы данных – интенсивность (различные спектральные слои), температурные данные, маски облаков и водоемов.

Для того, чтобы в процессе композиции не произошло смешения различных типов данных, каждый слой имеет свой тип (температура, интенсивность, маска и др.), которые задаются в момент создания слоя. Это позволяет контролировать типы слоев, которые даются на вход композиции слоев. Объект-функция предоставляет информацию о том, какого типа должен быть каждый слой на входе композиции, и не принимает слои, которые не соответствуют нужному типу.

Кроме типа, слой также характеризуется типом изображения, которое в нем хранится – grayscale (серое) или RGB, глубиной цвета – от 8 до 16 бит на пиксель.

## 5. ЗАКЛЮЧЕНИЕ

Разработана библиотека классов C++ для хранения и обновления изображений участки земной поверхности полу-

ченных со спутника. Библиотека дает возможность переводить полученные изображения в географические проекции и совмещать их с уже существующими изображениями с помощью объектов-функций. На рис. 6 показан пример композиции нескольких сеансов с использованием простейшей функции композиции – наложения с прозрачностью 50%.

В библиотеке большое внимание уделено надежности хранения данных. При модификации данных результаты помещаются во временный слой, и только после того, как результаты признаны удовлетворительными производится замещение основного слоя карты.

Пример использования библиотеки, осуществляющий проектирование:

```
// инициализация спутникового сеанса
AVHRRpassport seance;
seance.Init("e:\\mapman\\D43331.psp");
// загрузка входного изображения
auto_ptr<Almage> inputImage(
    ::AlmageLoad(0,true,"./mapman/d43331_4.tif"));
// открытие библиотеки карт
GeoMap::MapLibrary mapLibrary("/mapman/sampleLib/");
// получение объекта для работы с картой
GeoMap::MapManager& mapManager =
    mapLibrary.getMapManager("MapStringID");
// проверка, попадает ли сеанс на карту и проектирование
int layered;
if(mapManager.isIntersect(seance))
    layered = mapManager.project(
        "id","name","INTENSITY","desc",
        inputImage.get(),seance);
else
    return;
// композиция с существующим слоем с id = 0
GeoMap::ComposeMethodTransparent composeMethod(0.5)
composeMethod.setInputLayerId(0,0);
composeMethod.setInputLayerId(1,layered);
int compId = mapManager.compose(0,composeMethod);
// прием предварительного слоя в качестве основного
mapManager.accept(0,compId);
// удаление временного слоя с проекцией
mapManager.deleteLayer(layered);
```

В настоящее время библиотека находится на этапе внедрения в аппаратно-программный комплекс для приема и тематической обработки спутниковых сеансов AlcorSat, разрабатываемой НПП ОПТЭКС.

## 6. ЛИТЕРАТУРА

- [1] Nepomnyaschy P.V., Yurin D.V. Satellite Images Transformation to Geographical Projection with Adaptive Choice of Optimal Approximation Method According to Local Compression Coefficient. // In Conference Proceedings. 12-th International Conference on Computer Graphics GraphiCon'2002 –P. 299–304. Nizhny Novgorod, September 16-21, 2002.
- [2] Соловьёв А.Д. Математическая картография. – изд. Недра, Москва, 1969.
- [3] G. Goodrum, K.B. Kidwell, W. Winston, NOAA KLM Users Guide, 1999.

[4] Прэтт У. Цифровая обработка изображений: Пер. с англ. – М.: Мир, 1982. – Кн.1 – 312 с., ил., Кн.2 – 480 с., ил.

[5] Ярославский Л.П. Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику. – М. Радио и связь, 1987. – 296с.:ил.

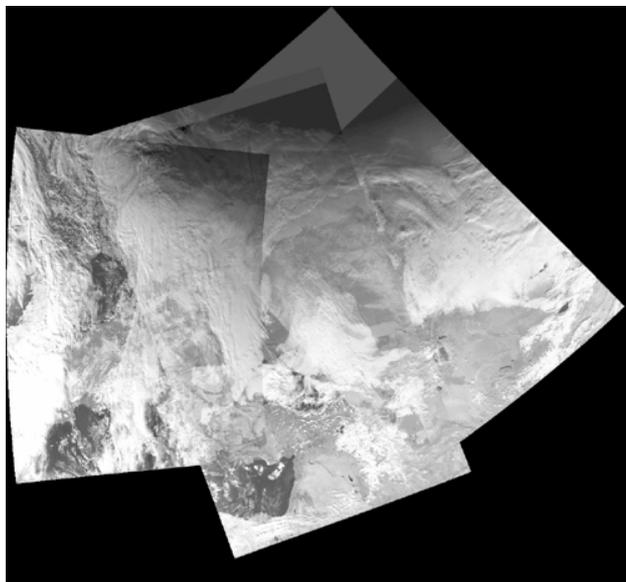


Рисунок 6. Результат композиции трех сеансов (наложение друг на друга с прозрачностью)

## Об авторах

Непомнящий Павел Владимирович, студент 6го курса Московского Физико-Технического Института, факультет Физической и Квантовой Электроники, кафедра Центр Открытых Систем и Высоких Технологий.

E-mail: npv@cos.ru

Юрин Дмитрий Владимирович, кфмн.

E-mail: yurin@cos.ru