

# Автоматическая оценка облачности по методике ACCA в QGIS

[Обсудить в форуме](#) Комментариев — 47

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/ggis-acca.html>

Описание расширения QGIS позволяющего использовать алгоритм автоматической оценки облачности

## Содержание

- [1 Получение и установка](#)
- [2 Интерфейс плагина](#)
- [3 Внутреннее устройство плагина](#)
  - [3.1 Архитектура плагина](#)
  - [3.2 Реализация](#)
    - [3.2.1 Алгоритм преобразования данных LANDSAT](#)
    - [3.2.2 Алгоритм ACCA](#)
- [4 Ссылки](#)

## Получение и установка

Расширение доступно из [официального репозитория](#).

Исходный код можно взять на GitHub: [https://github.com/sergey-bastrakov/Acca\\_Plugin](https://github.com/sergey-bastrakov/Acca_Plugin)

Установка модулей подробно описана в статье [Установка модулей расширения в QGIS](#).

## Интерфейс плагина

Весь интерфейс представляет собой две формы, первая позволяет указать параметры алгоритма, вторая выводит прогресс выполнения.

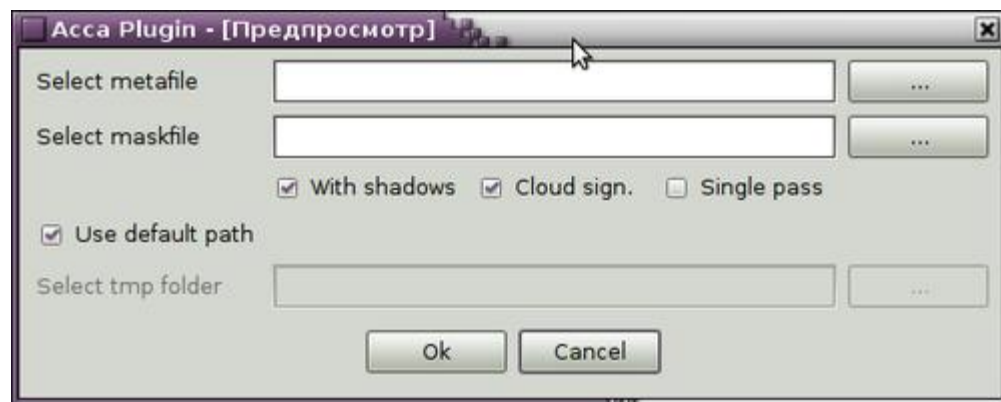


Рис. 1. Главная форма

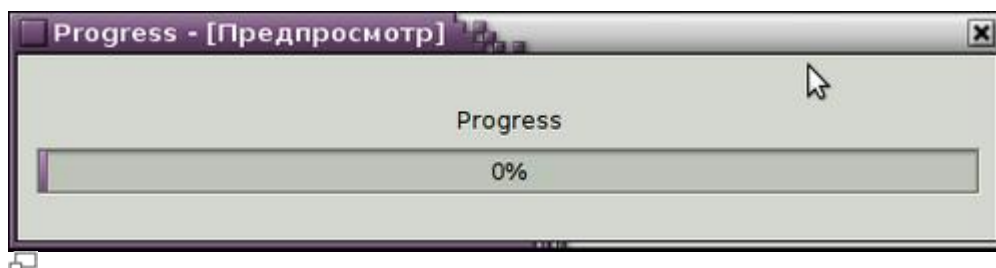


Рис. 2. Прогресс выполнения

После установки плагина в Quantum GIS, в меню появится пункт, вызывающий главное окно модуля. В появившемся главном окне необходимо указать два обязательных параметра:

- Путь до метафайла данных LANDSAT
- Путь до места, где будет сохраняться маска

Остальные параметры можно оставить по-умолчанию:

- Выполнять ли поиск облаков
- Использовать ли данные гистограммы по тепловому слою
- Выполнять ли один шаг
- Путь до преобразованных данных (По-умолчанию берется путь до папки с временными файлами вашей ОС)

После запуска алгоритма будет выведен прогресс выполнения, после чего программа завершается.

## Внутреннее устройство плагина

### Архитектура плагина

Функционально плагин можно разбить на несколько частей:

- TOAR – (англ. top-of-atmosphere radiance or reflectance and temperature) часть, которая вычисляет спектральную отражательную способность.
- ACCA – (англ. automatic cloud cover assessment) часть, которая содержит непосредственно алгоритм поиска облаков и построения маски облачности.
- Модуль для Quantum GIS – часть, которая обеспечит взаимодействие ГИС с алгоритмом ACCA
- Прочее – к этой части можно отнести различные модули, которые не попадают в вышеперечисленные категории. Например, файлы, которые содержат графический интерфейс, файлы содержащие описание модуля и так далее.

Блок-схема модуля:

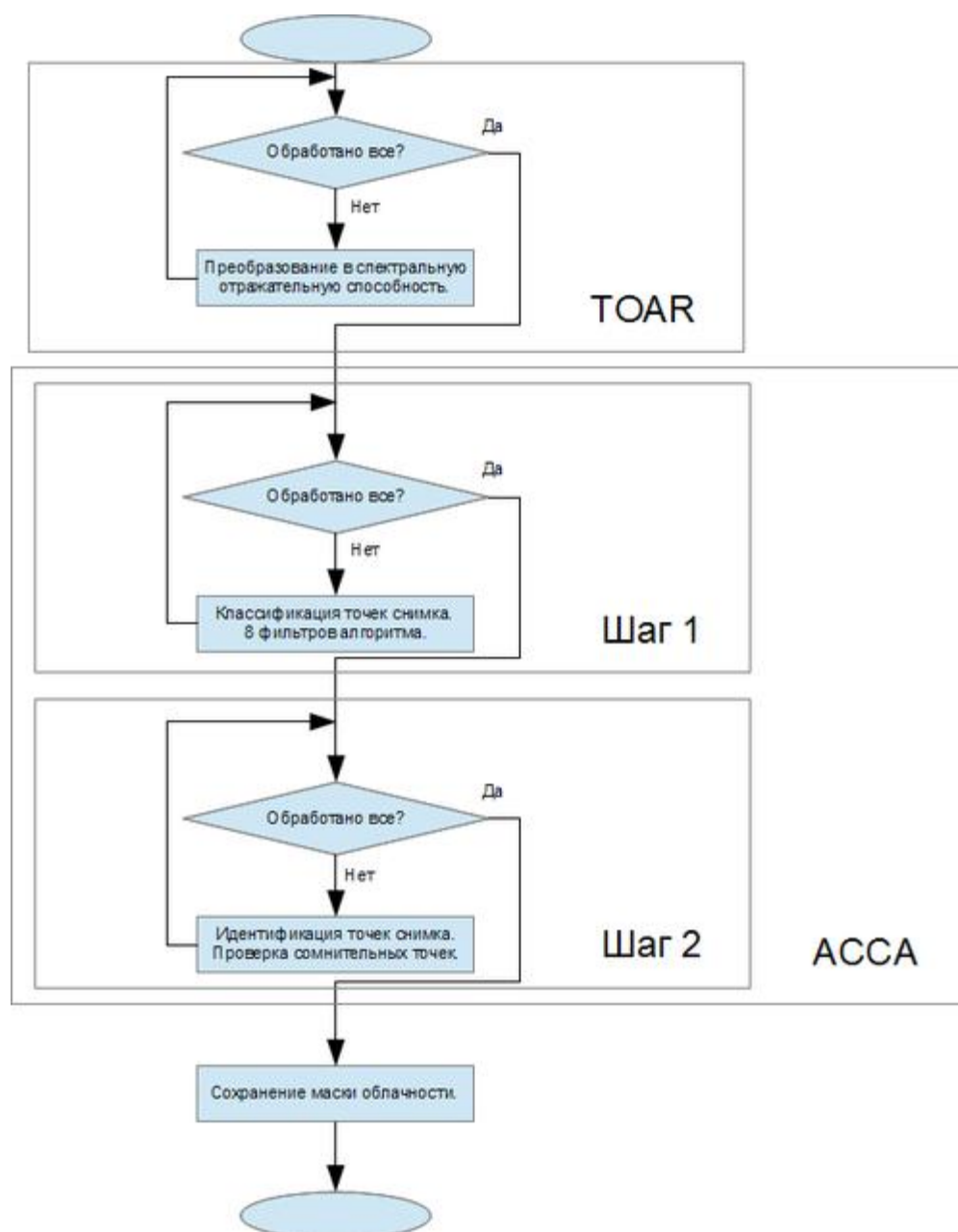


Рис 3. Блок-схема модуля

## Реализация

### Алгоритм преобразования данных LANDSAT

Для решения задачи преобразования данных LANDSAT был написан класс `SToar`, который включает в себя все необходимое для работы над снимком.

Снимок открывается с помощью библиотеки `Gdal`, она же предоставляет интерфейс для работы со слоями. Снимок в программе представляет собой набор двумерных массивов чисел – от 0 до 255 для каждого слоя. Каждый канал обрабатывается отдельно. С целью ускорения обработки канал разбивается на блоки 2000 на 2000 точек. Блоки обрабатываются сверху вниз слева направо.

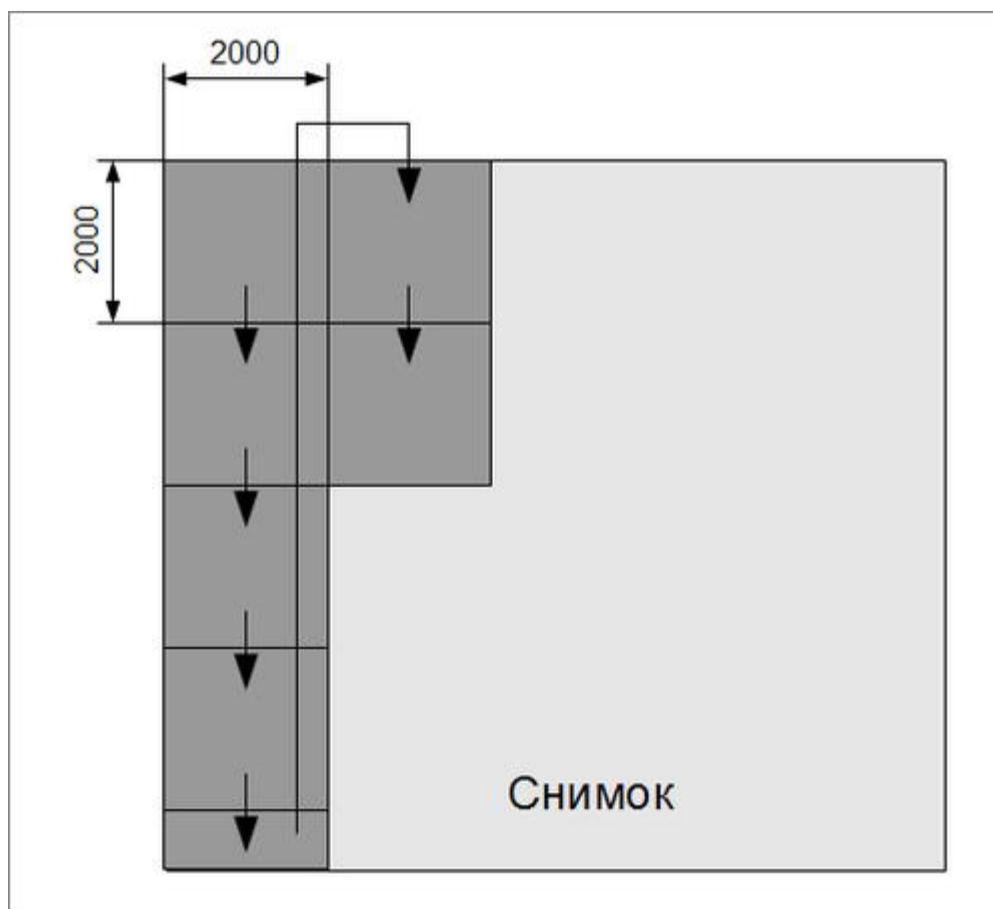


Рис. 4. Схема обработки снимка

Каждый блок обрабатывается целиком, библиотека NumPy позволяет выполнять поэлементные операции над массивами.

В зависимости от обрабатываемого слоя к снимку применяется свой набор операций. Сначала для каждого слоя рассчитывается плотность потока у датчика по формуле (1), далее для 2-5 каналов считается спектральная отражательная способность (2), а для теплового слоя температура (3).

Результат сохраняется в формате входных данных, кроме типа значений, у выходных данных тип float32.

UML диаграмма класса CToar:

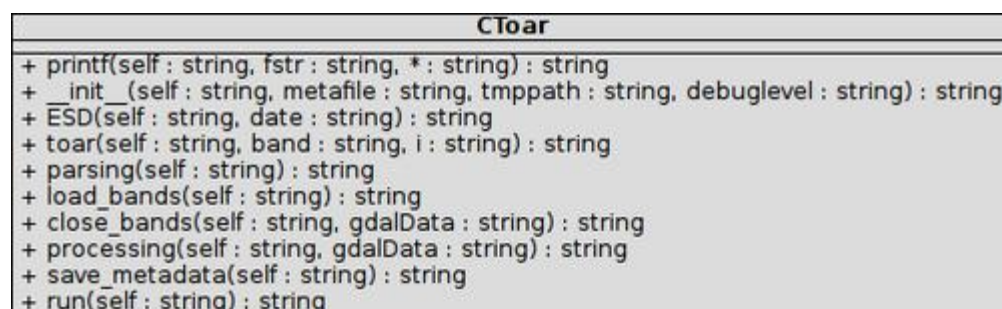


Рис. 5. UML диаграмма класса CToar

Обработка снимка происходит в функции processing, которая выполняется в своем потоке. Непосредственно вычисления производятся в функции toar.

Кроме обработки данных LANDSAT, необходимо произвести разбор метаданных и сохранить нужные для дальнейшей работы с алгоритмом ACCA. Разбор данных осуществляет функция parsing, а сохраняет нужные – save\_metadata.

Процесс обработки снимков достаточно длительный, поэтому предусмотрен отладочный вывод, который становится активным при debuglevel=1. Так же в процессе обработки экземпляр класса CToar испускает сигнал

progress, который может передавать информацию о прогрессе выполнения.

## Алгоритм АССА

Алгоритм содержится в классе САсса. В состав класса входит весь необходимый для решения задачи функционал.

Обработка снимка происходит так же, как и в классе СТоар. Снимок разбивается на блоки 2000 на 2000 точек. Но все каналы обрабатываются вместе. Так как в фильтрах проверяются значения из различных каналов.

При реализации класса САсса основной проблемой являлась проверка каждой точки, проходит она фильтр или нет. Поточечная обработка заняла бы слишком много времени. Поэтому было принято решение использовать маски - двумерный массив булевых элементов. Для каждого фильтра строилась такая маска. Если значения элемента в маске True, то соответствующая точка снимка фильтр проходит. После каждого фильтра маска умножается на маску с предыдущего шага. В результате те точки, которые прошли все фильтры, будут иметь True в соответствующей точке маски. Классификация точек производится при помощи функции NumPy – putmask. Эта функция заменяет значения точек в маске облачности, соответствующих значениям True в маске фильтра, на заданное значение.

В процессе обработки снимка на первом шаге ведется статистика различных классов точек. На её основе делается вывод, что нужно сделать на втором шаге. Например, строится гистограмма для точек температурного слоя. Из неё берутся пороговые температуры для второго шага.

Процесс вычисления получился ресурсоемким, для запуска алгоритма требуется минимум 1ГБ оперативной памяти и порядка 1,5ГБ на жестком диске. От процессора зависит только скорость расчетов. На 4-х ядерном процессоре AMD A6 – 1,5 ГГц время выполнения составляет порядка 10 минут.

UML-диаграмма класса САсса:

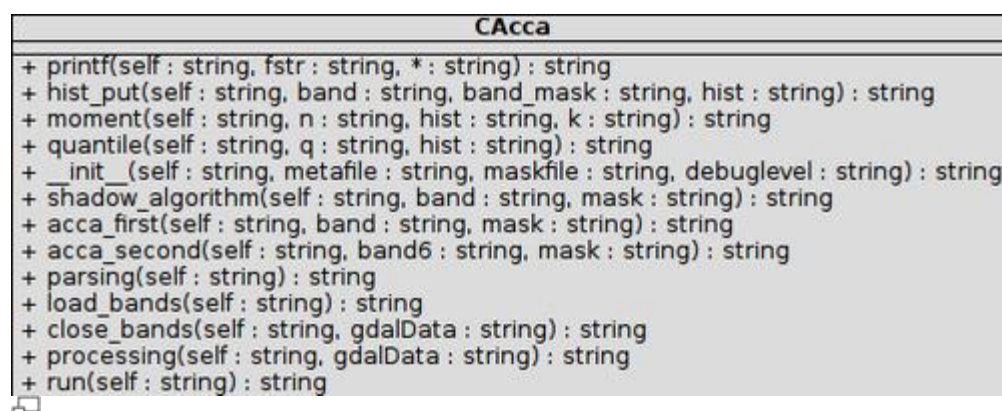


Рис. 6. UML диаграмма класса САсса

Точно так же как и в классе СТоар функция processing, которая производит обработку снимка, выполняется в отдельном потоке. Здесь так же происходит вывод прогресса, если debuglevel=1 и точно так же испускается сигнал progress.

Результат работы алгоритма:



Рис. 7. Снимок



Рис. 8. Итоговая маска, отредактированная для лучшего отображения

## Ссылки

1. [Описание алгоритма АССА](#)

[Обсудить в форуме](#) Комментариев — 47

Последнее обновление: 2014-05-15 00:12

Дата создания: 01,07,2012

Автор(ы): [Сергей Бастраков](#)