

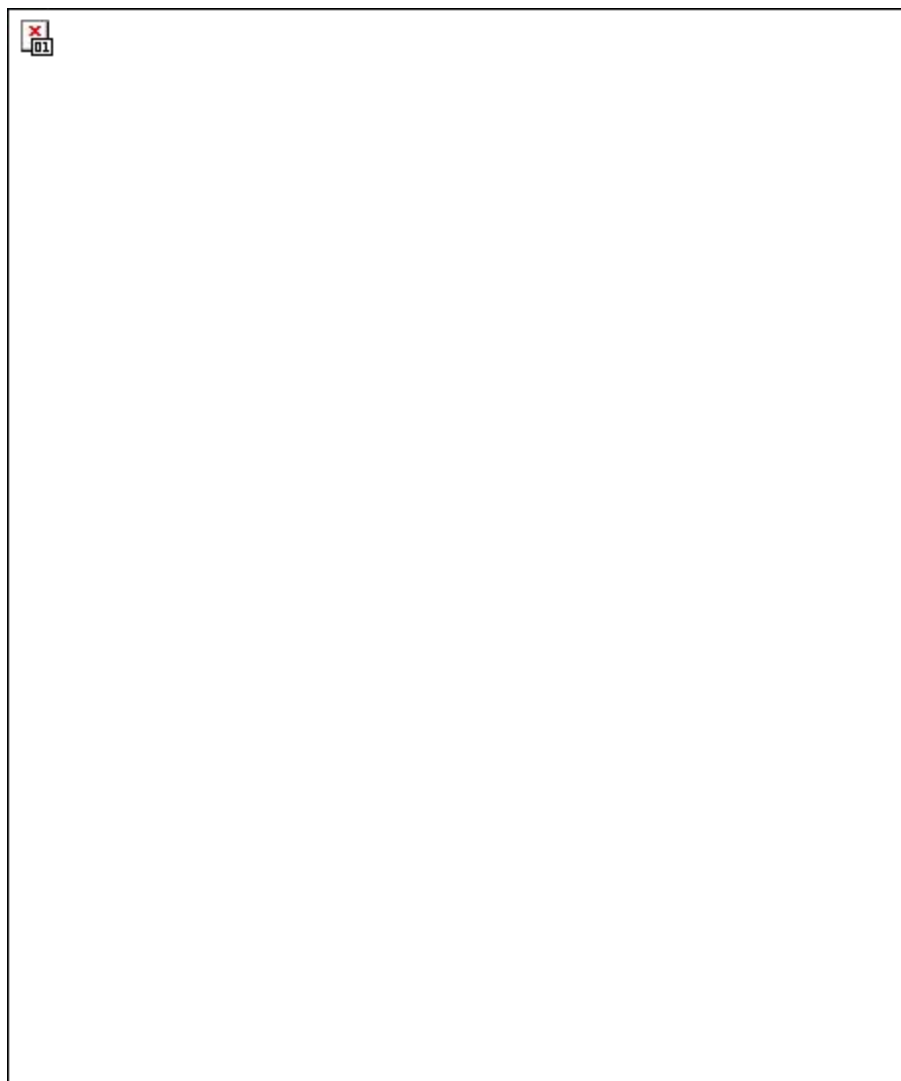
- [Главная](#)
- [Вопросы и ответы](#)

Создание каталога данных дистанционного зондирования CBERS

Описание и получение данных

[Обсудить в форуме](#) Комментариев — 2

Каталог космических снимков CBERS-2 (China-Brazil Earth Resources Satellite-2) доступен исключительно онлайн по адресу <http://www.dgi.inpe.br/CDSR/>, на португальском и английском языках. На спутнике [CBERS](#) установлены три камеры ДЗЗ: [WFI](#), [CCD](#), [IRMSS](#).



Загружаемая (оффлайн) версия каталога, которая незаменима при отборе большого количества снимков - отсутствует. Целью данной статьи является пример того, как можно создать такой каталог, используя несложные манипуляции с его онлайн-версией. Статья иллюстрирует процесс для данных полученных сенсором CCD. Для WFI и IRMSS процедура аналогична.

Для создания каталога нам нужны:

1. полигоны с границами, соответствующими отдельным сценам;
2. атрибутивная информация к каждой сцене;
3. превью (по желанию)
можно или скачать их все (что потребует немало трафика) и впоследствии привязать, получив полностью автономную версию каталога, или создать в атрибутивной таблице поле с гиперссылкой на

превью на официальном сайте.



Получение информации о сцене

Чтобы узнать, какую информацию о снимках можно получить, рассмотрим описание любой сцены в онлайн-каталоге. Каждая сцена имеет свой уникальный идентификатор, для того, чтобы посмотреть сцену с ID = CB2CCD14912420070112 можно воспользоваться следующей ссылкой:

<http://www.dgi.inpe.br/CDSR/manage.php?INDICE=CB2CCD14912420070112&DONTSHOW=1>



| | | |
|-------------|------------------|------------|
| спутник | Satelite | CB2 |
| сенсор | Sensor | CCD |
| пролет | Orbita | 149 |
| ряд | Ponto | 124 |
| дата съемки | Data de Passagem | 2007-01-12 |
| виток | Revolucao | 16920 |
| координаты | Latitude Norte | -20.90260 |

| | | |
|--|----------------------|----------------|
| (широта/долгота) верхнего правого угла снимка | Longitude Oeste | -41.65050 |
| координаты | Latitude Sul | -22.06050 |
| (широта/долгота) левого нижнего угла снимка | Longitude Leste | -40.79250 |
| время съемки по гринвичу | Tempo Central(GMT) | 12:38:23 |
| ориентация снимка | Orientacao da Imagem | 8.50435 |
| надир | Angulo Nadir | 25.7263 |
| угол солнца | Azimuth Sol | 96.7175 |
| высота солнца | Elevacao do Sol | 59.2737 |
| облачность | Q1, Q2, Q3, Q4 | 10, 10, 10, 10 |



Получение ID сцены

Из вышеприведенной ссылки понятно, что зная ID всех нужных снимков, можно скачать страницы с метаданными и превью. Итак, как получить список всех валидных ID?

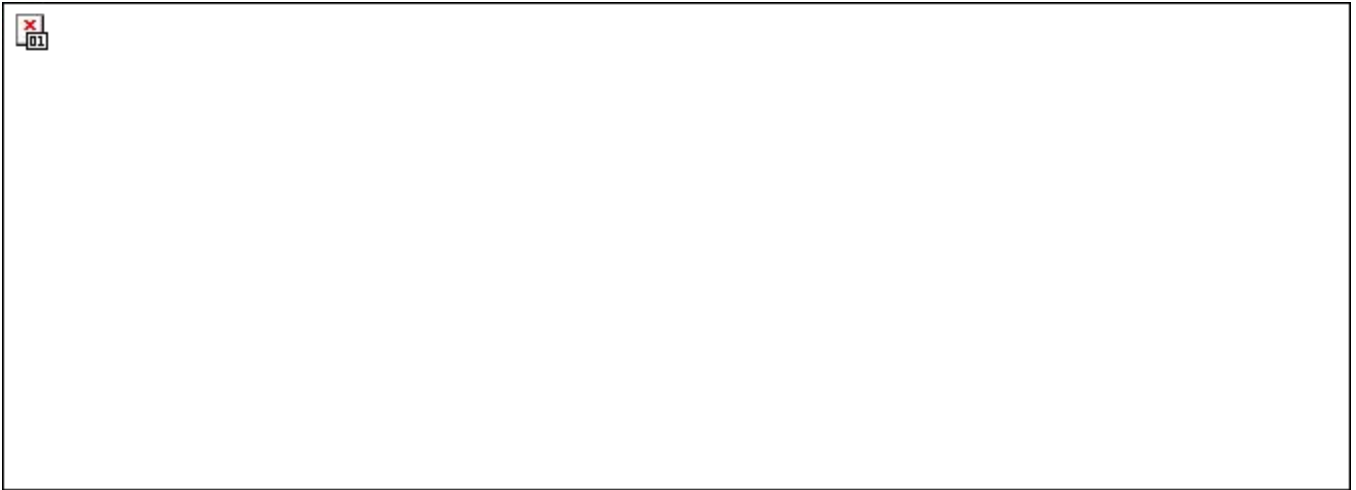
Можно решить задачу "в лоб", простым перебором по шаблону CB2CCD<path><row>YYYYMMDD, разделяя корректные и ложные ID по размеру возвращаемой сервером страницы. Но число потенциальных комбинаций слишком велико. Можно, используя данные о периодичности пролета по одному и тому же витку и время последнего пролета над заданной территорией, отфильтровать заведомо некорректные комбинации. Но тот факт что спутник в определенный день пролетал над заданной местностью еще не означает, что соответствующий снимок был сделан, сохранен и присутствует в каталоге.

Другой способ - запросить список ID у сервера, задав нужные критерии поиска. Например, все снимки, сделанные с 1 января 2006 по 1 января 2007 сенсором CCD для территории лежащей между 0N - 10N и 70W – 90W.



Как видно из результата, сервер вернул первую тысячу из N найденных снимков (50 страниц по 20 снимков на каждой (Mostradas apenas as 1000 primeiras imagens - refine sua consulta na ausencia da cena de interesse). Нас интересует запрос, который браузер послал серверу. Т.к. каталог сверстан с применением фреймов - пользователь не видит ссылки в адресной строке браузера, для того чтобы узнать какой именно запрос был послан нужно использовать любой анализатор HTTP трафика (автор использовал встроенный в Proxomitron,

<http://www.proxomitron.info>).



Браузер, посылая запрос "GET <url> HTTP/1.0", получает от сервера в ответ страницы с выборкой из каталога. Редактируя передаваемые параметры, можно запросить выборку снимков с произвольными (в пределах здравого смысла) критериями. Огорчает только, что она ограничена одной тысячей снимков.

```
SELECT * FROM Scene WHERE Deleted=0 AND CloudCoverMethod='M' AND Satellite ='CB2' AND
Sensor ='CCD' AND CenterLatitude <= 10. AND CenterLongitude >= -90. AND
CenterLongitude <= -70 AND Date>='2006-01-1 00:00:00' AND Date<='2007-01-31 23:59:59'
ORDER BY Date DESC, Path, Row LIMIT 1000
```

Это выражение - SQL запрос к базе данных снимков. В строке браузера он выглядит так:

```
http://www.dgi.inpe.br/CDSR/mantab.php?TAM=G&pagina=1&sql=SELECT%20*%20FROM%20Scene%20W
HERE%20Deleted=0%20AND%20CloudCoverMethod='M'%20AND%20%20%20Satellite%20='CB2'%20AND%20
%20%20Sensor%20='CCD'%20AND%20%20%20CenterLatitude%20%3C=%2010.%20AND%20%20%20CenterLon
gitude%20%3E=%20-90.%20AND%20%20%20CenterLongitude%20%3C=%20-
70%20AND%20%20%20Date%3E='2006-01-1%2000:00:00'%20AND%20%20%20Date%3C='2007-01-
31%2023:59:59'%20ORDER%20BY%20Date%20DESC, Path, Row%20LIMIT%201000
```

Наиболее интересны для нас следующие параметры:

- pagina=1 позволяет выбрать нужную страницу с результатами запроса;
- Date>='2006-01-1 00:00:00'
- Date<='2007-01-31 23:59:59' позволяют разбить запрос на несколько временных интервалов;
- LIMIT 1000 позволяет снять ограничение в 1000 сцен, возвращаемых при поиске, указав число, заведомо большее чем количество снимков, например 1000000. Конечно, делать это стоит только совместно с указанием узких временных рамок, поскольку в противном случае на сервер при таком запросе создается очень большая нагрузка;
- ORDER BY Date DESC, Path, Row позволяет отсортировать выдачу сцен по нескольким критериям.

Анализируя код далее, мы видим уже знакомый нам формат ссылки для запроса страницы с подробной информацией по выбранной сцене manage.php?INDICE=CB2CCD18511420070108&DONTSHOW=1.



Получение списка ссылок

Теперь нам известно все, что нужно для получения полного списка ID. Чтобы не создавать ссылок руками и ускорить обработку скаченных страниц, применим perl. Первым делом пишем простой скрипт для генерирования ссылок на страницы каталога с выборкой. Будем генерить пакеты ссылок на все снимки, сделанные за месяц.

```
#!/usr/local/bin/perl
```

```
#задаем переменные, определяющие начальную и конечную даты и количество страниц
#ради упрощения скрипта будем указывать их непосредственно в его коде
$datel = "2006-12-1%2000:00:00";
```

```

$date2 = "2006-12-31%2023:59:59";
$lastpage = "73";

#открываем файл, куда будем записывать сгенерированные ссылки
open (DL1, ">_dl1_list.ion");

$p = 1;

#запускаем цикл, в котором создаются ссылки с параметрами, указанными в переменных, и
записываем их в файл формата *.ion
while ($p<=$lastpage) {
    print DL1 "$p\htm
http:\/\/www\dgi\inpe.br\CDSR\/mantab.php?TAM=G&pagina={$p}&sql=select\%20*\%20
FROM\%20Scene\%20WHERE\%20Deleted=0\%20AND\%20CloudCoverMethod=\ 'm'\%20AND\%20\%20\%
20Satellite\%20=\ 'CB2'\%20AND\%20\%20\%20CenterLatitude\%20\%3E=\%20-
90\.\%20AND\%20\%20\%20CenterLatitude\%20\%3C=\%2090\.\%20AND\%20\%20\%20CenterLongitu
de\%20\%3E=\%20-
90\.\%20AND\%20\%20\%20CenterLongitude\%20\%3C=\%2090\.\%20AND\%20\%20\%20Date\%3E=\ '
$date1'\%20AND\%20\%20\%20Date\%3C=\ '$date2'\%20ORDER\%20BY\%20Date\%20DESC\,Path\,R
ow\%20LIMIT\%2050000\n";
    $p++;
}

close DL1;

```

Т.к. все страницы (в данном случае все 73) по умолчанию при сохранении имели бы одинаковое имя, мы создали файл формата ion, понимаемый ReGet'ом. Формат его <имя файла><пробел><линк на файл>. Использование его позволяет присвоить каждой скаченной странице заданное автоматически из переменной \$p имя. Сгенерировав файл со ссылками, импортируем в ReGet и скачиваем. Получаем множество файлов html, из которых нам нужно извлечь ID снимков, а лучше сразу фрагмент ссылки для скачивания описания сцен.

Пишем скрипт, который ищет ID по шаблону:

```

#!/usr/local/bin/perl
#создаем список скаченных страниц
@list=glob("*\htm");
#в цикле ищем ID по шаблону и все найденные соответствия помещаем в массив @url
foreach $list (@list) {
    open (TMP, "$list");
    @tmp = <TMP>;
    foreach $tmp (@tmp) {
        if ($tmp =~ /manage\.php\?INDICE\=CB2CCD0-9+\&DONTSHOW\=1/) {
            push @url, ("${&}");
        }
    }
}
#открываем файл для записи списка ID
open (OUT, ">_list1.txt");
#создаем хеш и отбирая с его помощью уникальные ID, выкидываем их в файл
%tmp = ();
foreach $item (@url) {
    $item =~ s/manage\.php\?INDICE\=//;
    $item =~ s/\&DONTSHOW\=1//;
    print OUT "$item\n" unless $tmp{$item}++;
}

```

Итак, мы получили желаемый список. Теперь нужно скачать страницы с метаданными ко всем снимкам и, по желанию, превью. Необходимый для загрузки список ссылок нам поможет сделать perl:

```

#!/usr/local/bin/perl
open (TMP, "_list1.txt");
@list = <TMP>;
open (DL2, ">_dl2_list.ion");

```

```
foreach $line (@list) {
$line =~ s/\n//;
print DL2 "$line\.meta
http://www.dgi.inpe.br/CDSR/manage.php?INDICE=$line&DONTSHOW=1\n";
print DL2 "$line\.jpg
http://www.dgi.inpe.br/CDSR/display.php?TABELA=Browse&PREFIXO=Cbers&INDICE=$line\n";
}
close TMP;
close DL2;
```

Скрипт использует полученный выше список ID и создает для каждого снимка две ссылки: для html страницы с метаинформацией и для превью. Сохраняются они под именами *.meta и *.jpg. Полученный список опять загружаем в ReGet и запускаем на загрузку. Внимание! Прежде чем закидывать все полученные ссылки, посмотрите что закидывается по первым. Если вы ошиблись при генерации ссылок (или на сервере что-то поменялось, и скрипты генерируют невалидные ссылки), то вместо страницы с мета и превью вы будете получать страницу с ошибкой, а администратор сервера - оседающие в логах тысячи сообщений об ошибках. Закончиться это может как блокированием вашего IP на сервере, так и изменением структуры запросов на сервере.



Создание каталога

Начнем с извлечения информации о снимках из html файлов, ведь координаты углов нам будут нужны для построения полигонов. Используем тот факт, что все html выглядят однотипно, и будущие значения полей атрибутивной таблицы занимают строки с одним и тем же номером во всех html файлах (Примененный ниже способ извлечения данных из текстовых файлов, а html по сути текстовый файл, не универсален, мы используем его только ради быстроты).

```
#!/usr/local/bin/perl
@metafiles = glob("*.meta");
#открываем файл и записываем в него заголовки полей атрибутивной таблицы, закрываем
его. Если файл уже существует его содержимое будет потеряно
open (META2, ">_meta2.txt");
print META2
"id,sitellite,sensor,path,row,date,time,revolution,orientation,nadir,azimuth,rise,q1,q2
,q3,q4,ULY,ULX,LRY,LRX\n";
close META2;
#открываем цикл, в котором будем построчно обрабатывать все метафайлы
foreach $file (@metafiles) {
$file =~ s/\n//;
open (META1, "$file");
@meta = <META1>;
close META1;

open (META2, ">>_meta2.txt");

#очищаем от "мусора" нужные нам значения
foreach (@meta) {
$_ =~ s/<td>>//;
$_ =~ s/<\/td>>//;
$_ =~ s/<td align=\"center\" nowrap>Q1\&nbsp; //;
$_ =~ s/<td align=\"center\" nowrap>Q2\&nbsp; //;
$_ =~ s/<td align=\"center\" nowrap>Q3\&nbsp; //;
$_ =~ s/<td align=\"center\" nowrap>Q4\&nbsp; //;
$_ =~ s/\n//;
}

$file =~ s/\n//;

#добавляем в таблицу извлеченную атрибутивную информацию
print META2 (" $file,$meta61,$meta65,$meta69,$meta73,$meta77,
$meta101,$meta81,$meta105,$meta110,$meta114,$meta118,$meta124,$meta125,$meta128,
$meta129,$meta85,$meta89,$meta93,$meta97\n");
close META2;
}
```

В отдельном скрипте рассчитаем координаты углов снимка и создадим файл, автоматизирующий создание полигонов в ArcView.

```
#!/usr/local/bin/perl
@files = glob("*.txt");
open (POLY, ">poly.txt");
print POLY "id_cbers,UL_Long,UL_Lat,UR_Long,UR_Lat,LR_Long,LR_Lat,LL_Long,LL_Lat\n";
close POLY;
foreach $file (@files) {
$file =~ s/\.txt//;

open (META1, "<$file.txt");
@meta = <META1>;
close META1;

#удаляем первую строку из считанного файла (с названиями полей таблицы)
splice (@meta, 0, 1);

$sin = sin (3.14159*8.50435/180);
$cos = cos (3.14159*8.50435/180);

foreach (@meta) {
$_ =~ s/\n//;
@a = split(/\\/,/,$_);

#получаем координаты верхнего левого и правого нижнего углов снимка (или экстента)
$id = $a0;
$ULY = $a16;
$ULX = $a17;
$LRX = $a18;
$LRX = $a19;

#рассчитываем координаты четырех углов снимка (пояснения ниже в тексте)
$x = sqrt (($LRX - $ULX)^2 + ($ULY - $LRX)^2);
$cULX = $ULX;
$cULY = $ULY;
$cLRX = $LRX;
$cLRY = $LRY;
$cLLX = $ULX - $x*$sin;
$cLLY = $ULY - $x*$cos;
$cURX = $LRX + $x*$sin;
$cURY = $LRY + $x*$cos;

#добавляем координаты углов в файл
open (POLY, ">>poly.txt");
print POLY "$id,$cULX,$cULY,$cURX,$cURY,$cLRX,$cLRY,$cLLX,$cLLY\n";
close POLY;
}
}
```

Расчет координат углов полигонов производится исходя из имеющихся данных: координат верхнего левого и нижнего правого углов и угла наклона орбиты спутника по отношению к экватору. Строго говоря, неизвестно, координаты ли это углов снимка или углов экстента (прямоугольника, в который вписан снимок). Будем пока придерживаться первого предположения.

Координаты в градусах, а расчеты мы ведем для прямоугольных координат, потому результаты грубы. Учитывая тот факт, что размеры снимков невелики, погрешность приемлива.

Соотношение сторон снимка 1:1. Используя тригонометрические функции, рассчитаем оставшиеся координаты как результат смещения исходных точек на $x * \sin a$ (для долготы) и $x * \cos a$ (для широты), где: x - расстояние между верхним левым и нижним правым углами снимка (диагональ квадрата полигона), ее мы вычисляем из теоремы Пифагора ($x = \sqrt{((LRX - ULX)^2 + (ULY - LRY)^2)}$)
 a - угол наклона орбиты спутника к экватору в радианах.
Тригонометрические функции \sin и \cos встроены в perl, что облегчает нашу задачу.

Для нижнего левого угла:

$$cLLX = ULX - x * \sin a$$

$$cLLY = ULY - x * \cos a$$

Для верхнего правого угла:

$$cURX = LRX + x * \sin a$$

$$cURY = LRY + x * \cos a$$

Построив полигоны (любым известным способом), остается только присоединить атрибутику и проверить, корректен ли наш каталог. Для этого подгрузим любой доступный снимок (например, Landsat ETM+) и откроем страницу с превью соответствующего снимка в онлайн-каталоге.

Видно, что границы полигонов и превью почти совпадают. Отчетливо видно искажения полигонов (получились скошенные четырехугольники), причина видимо в расчете координат в градусах как прямоугольных.

В случае необходимости, зная формат ссылок на страницы с превью и информацией о сценах, мы можем создать дополнительные поля в атрибутивной таблице со ссылками на онлайн-каталог. В сгенерированном каталоге имеется поле "preview", по хотлинку с него можно непосредственно открывать превьюш или делать список для загрузки.

В процессе создания каталога выяснилось, что у некоторых сцен координаты углов указаны неверно, вот их список:

CB2CCD21106720040207

CB2CCD21106820040207

CB2CCD21206420040925

CB2CCD21206520040925

CB2CCD02506620040925

CB2CCD02506720040925

CB2CCD02506920040207

CB2CCD02507020040207

CB2CCD266620031214



Готовый каталог

Если по той или иной причине невозможно воспользоваться этой инструкцией, можно скачать готовую версию [финального каталога](#) актуального на 27 января 2007.

[Обсудить в форуме](#) Комментариев — 2

Последнее обновление: March 01 2011

Дата создания: 15.04.2007

Автор(ы): [Александр Маниша](#)