

# Cameras

## Importer les données avec OSM2PGSQL

```
docker run --name postgis -e POSTGRES_PASSWORD=password -p 5432:5432 -d postgis/postgis
```

```
local cameras = osm2pgsql.define_node_table('cameras', {
  { column = 'id', sql_type = 'serial', create_only = true },
  { column = 'geom', type = 'point' },
})

local highways = osm2pgsql.define_way_table('highways', {
  { column = 'id', sql_type = 'serial', create_only = true },
  { column = 'geom', type = 'linestring' },
})

local buildings = osm2pgsql.define_area_table('buildings', {
  { column = 'id', sql_type = 'serial', create_only = true },
  { column = 'geom', type = 'polygon' },
})

function osm2pgsql.process_node(object)
  if object.tags.man_made == 'surveillance' then
    cameras:insert({
      geom = object:as_point( )
    })
  end
end

function osm2pgsql.process_way(object)
  if object.is_closed and object.tags.building then
    buildings:insert({
      geom = object:as_polygon()
    })
  end
end
```

```

    })
end

if object.tags.highway then
    highways:insert({
        geom = object:as_linestring()
    })
end
end

function osm2pgsql.process_relation(object)
    if object.tags.type == 'multipolygon' and object.tags.building then
        local geom = object:as_multipolygon()

        for g in geom:geometries() do
            buildings:insert({
                geom = g,
            })
        end
    end
end
end
end

```

```
osm2pgsql -d toto -U toto -W -H localhost -O flex -S extract_features.lua st_quentin.osm.pbf
```

## Inporter les données de sous-surveillance.net

<https://toulouse.sous-surveillance.net/spip.php?page=cameras&format=json&details=2&lang=fr>

```
sed -i -e 's/id_camera/node_id/g' camera.json
```

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgres user=postgres password=password host='localhost'"
"camera.json" -nln cameras -append -t_srs "EPSG:3857"
```

## Utiliser postgis

### Calculer une distance

```
with c1 as (select geom as g from cameras where id = 1), c2 as (select geom as g from cameras
where id = 2) select ST_Distance(c1.g, c2.g) from c1, c2 ;
```

## calculer les ways qui sont a moins de 100m d'une camera

```
select c.node_id, h.way_id from cameras c left join highways h on ST_DWithin(c.geom, h.geom,
100) where c.id = 1
```

## Calculer les ways visible d'une camera

```
create view visible_streets_from_cam as
with
  fields as (
    select
      c.node_id as camera_id,
      h.way_id as street_id,
      -- Changer ici pour la résolution du test d'intersection
      ST_Segmentize(h.geom, 2) as street_geom,
      c.geom as camera_coord
    from
      cameras c
    left join
      -- Changer ici pour la distance à la caméra
      highways h on ST_DWithin(c.geom, h.geom, 100)
    --where c.node_id = 9760071131
  ),
  segments as (
    select
      camera_id,
      street_id,
      ST_MakeLine(camera_coord,p) as seg_line,
      seg_id
    from (
      select
        fields.camera_id,
        fields.street_id,
        fields.camera_coord,
```

```

        generate_series(1, ST_NPoints(fields.street_geom)) as seg_id,
        ST_PointN(fields.street_geom, generate_series(1,
ST_NPoints(fields.street_geom))) as p
    from fields
    ) as s
),
line_of_sight as (
    select
        segments.seg_id,
        segments.camera_id,
        segments.street_id,
        buildings.area_id as building_id,
        ST_Intersects(seg_line, buildings.geom) as inter
    from segments
    left join buildings
    on ST_Intersects(seg_line, buildings.geom)
),
visible_street as (
    select
        camera_id,
        street_id,
        seg_id,
        not((inter is not null) and inter) as is_visible,
        building_id
    from line_of_sight
)
select
    distinct
        street_id,
        camera_id
from visible_street
where is_visible;

select STRING_AGG(distinct(street_id::text), ',') from visible_streets_from_cam;

```

```

psql -h localhost -U postgres -c "select STRING_AGG(distinct(street_id::text), ',') from
visible_streets_from_cam;" > ids.txt

```

# Editer du pbf avec osmium

Convertis en format opl (format textuelle, éditable)

```
osmium cat st_quentin.osm.pbf -f opl > st_quentin.opl
```

L'inverse

```
osmium cat st_quentin.opl -f pbf > st_quentin.osm.pbf
```

Avoir un diff

```
osmium diff st_quentin.osm.pbf merge.osm.pbf -f opl > diff.txt
cat diff.txt | grep "^+" # ajout
cat diff.txt | grep "^-" # suppression
```

Appliquer un fichier de changeset (attention écrase toute la relation ou tout le noeud)

```
osmium apply-changes st_quentin.osm.pbf osmChange.xml -o new.osm.pbf
```

## un peu de python...

```
import re

inputFile = "st_quentin.opl"
outputFile = "output.txt"

idsToEdit =
("w1087846855", "w1087920777", "w1087920778", "w1087920779", "w1087932398", "w1087942997", "w1087952
021", "w1087952022", "w1087952024", "w1087952025", "w1087960194", "w1087960196", "w1087960210", "w108
7960223", "w1087960224", "w1089446799", "w1089446800", "w1112075298", "w113961904", "w113961905", "w1
14536713", "w114666459", "w116382851", "w116382861", "w117355631", "w11773344", "w11773350", "w122006
038", "w126094356", "w126871656", "w126871657", "w126871658", "w126871664", "w126871669", "w126871675
", "w126871690", "w129705116", "w147848058", "w148736510", "w148825323", "w150550805", "w155480974", "
w15802834", "w15802836", "w165807675", "w168295763", "w170146623", "w170146625", "w172022081", "w1720
23915", "w172023916", "w173256365", "w173256368", "w173256369", "w173256729", "w173256730", "w1732567
31", "w173832779", "w174488908", "w177853034", "w177853035", "w180280918", "w180461913", "w180465548",
"w180468951", "w182628812", "w182629658", "w185057057", "w189637127", "w193375648", "w19844790", "w1
9847245", "w19848882", "w202737485", "w202970121", "w203030595", "w217421358", "w222393302", "w223415
```

```
74", "w22343527", "w22343529", "w22345126", "w225625930", "w22568170", "w22568182", "w22568375", "w22568378", "w22568386", "w22568388", "w22568424", "w22568488", "w22568523", "w22588043", "w22588046", "w22604437", "w22673740", "w22674191", "w22674884", "w22675194", "w22698778", "w227400308", "w227400309", "w23025712", "w277863607", "w28353561", "w29619599", "w29619604", "w315383203", "w31662553", "w31662878", "w31662879", "w31662880", "w31662964", "w31662965", "w31662968", "w31663124", "w330925134", "w330925135", "w330926195", "w331028474", "w331028475", "w339715316", "w339726718", "w339726719", "w346027918", "w349277858", "w349531415", "w35542048", "w363185344", "w364068225", "w365390727", "w370956707", "w38742540", "w38742541", "w40372065", "w40425070", "w40425074", "w4229251", "w443334075", "w443366920", "w447751613", "w447751614", "w48225928", "w492147752", "w49835957", "w49835959", "w49835960", "w50673023", "w509055318", "w509246741", "w509246744", "w509246745", "w509246746", "w509246747", "w509246753", "w51341167", "w51341168", "w515490602", "w51692324", "w52275569", "w52303054", "w52303060", "w525662191", "w525662193", "w52613839", "w52613863", "w526190596", "w530531180", "w530531181", "w530531182", "w531463493", "w531692522", "w531692526", "w531692527", "w536092843", "w536092847", "w553315235", "w587903986", "w587903987", "w587908623", "w587908627", "w587908630", "w587908634", "w59854247", "w628788987", "w630445718", "w630446538", "w630455616", "w636627411", "w636627412", "w641622109", "w641622111", "w641622112", "w641622114", "w641622115", "w641622116", "w641622118", "w650553749", "w662006651", "w662006652", "w662006654", "w694476166", "w698066552", "w705274809", "w705274810", "w705274811", "w708338741", "w764629037", "w770613905", "w770613907", "w829119358", "w829119361", "w829119362", "w829119363", "w829119365", "w829119366", "w829180626", "w829180633", "w829252460", "w85552224", "w87565425", "w879402866", "w880518011", "w881591456", "w881764531", "w881764532", "w881764537", "w881764538", "w881764539", "w881764540", "w881764543", "w881764545", "w881764546", "w881764547", "w881764549", "w881764552", "w88646884", "w88646885", "w88646897", "w89578796", "w89685774", "w89685814", "w89685825", "w930154230", "w930154231", "w944384994", "w963397962", "w963397963", "w991218943", "w991218944", "w991218945", "w991218946", "w991218947")
```

```
fo = open(outputFile, "w")
```

```
with open(inputFile) as f:
```

```
    for line in f:
```

```
        if line.startswith(idsToEdit):
```

```
            index = index + 1
```

```
            newLine = re.sub(r'( T)', r'\1camera=yes,', line)
```

```
            if line == newLine:
```

```
                print(line)
```

```
            fo.write(newLine)
```

```
        else:
```

```
            fo.write(line)
```

## Le même en c

(pas sur que ce soit plus opti)

```
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int StartsWith(const char *a, const char *b)
{
    if(strncmp(a, b, strlen(b)) == 0) return 1;
    return 0;
}

char *str_replace(char *orig, char *rep, char *with) {
    char *result; // the return string
    char *ins;    // the next insert point
    char *tmp;    // varies
    int len_rep; // length of rep (the string to remove)
    int len_with; // length of with (the string to replace rep with)
    int len_front; // distance between rep and end of last rep
    int count;    // number of replacements

    // sanity checks and initialization
    if (!orig || !rep)
        return NULL;
    len_rep = strlen(rep);
    if (len_rep == 0)
        return NULL; // empty rep causes infinite loop during count
    if (!with)
        with = "";
    len_with = strlen(with);

    // count the number of replacements needed
    ins = orig;
    for (count = 0; tmp = strstr(ins, rep); ++count) {
        ins = tmp + len_rep;
    }
}
```

```

tmp = result = malloc(strlen(orig) + (len_with - len_rep) * count + 1);

if (!result)
    return NULL;

// first time through the loop, all the variable are set correctly
// from here on,
// tmp points to the end of the result string
// ins points to the next occurrence of rep in orig
// orig points to the remainder of orig after "end of rep"
while (count--) {
    ins = strstr(orig, rep);
    len_front = ins - orig;
    tmp = strncpy(tmp, orig, len_front) + len_front;
    tmp = strcpy(tmp, with) + len_with;
    orig += len_front + len_rep; // move to next "end of rep"
}
strcpy(tmp, orig);
return result;
}

int main(void)
{
    FILE * input_file;
    FILE * output_file;
    char * line = NULL;
    char * new_line = NULL;
    size_t len = 0;
    ssize_t read;
    char ids[253][12] =
{"w1087846855", "w1087920777", "w1087920778", "w1087920779", "w1087932398", "w1087942997", "w1087952
021", "w1087952022", "w1087952024", "w1087952025", "w1087960194", "w1087960196", "w1087960210", "w108
7960223", "w1087960224", "w1089446799", "w1089446800", "w1112075298", "w113961904", "w113961905", "w1
14536713", "w114666459", "w116382851", "w116382861", "w117355631", "w11773344", "w11773350", "w122006
038", "w126094356", "w126871656", "w126871657", "w126871658", "w126871664", "w126871669", "w126871675
", "w126871690", "w129705116", "w147848058", "w148736510", "w148825323", "w150550805", "w155480974", "
w15802834", "w15802836", "w165807675", "w168295763", "w170146623", "w170146625", "w172022081", "w1720
23915", "w172023916", "w173256365", "w173256368", "w173256369", "w173256729", "w173256730", "w1732567

```

```
31", "w173832779", "w174488908", "w177853034", "w177853035", "w180280918", "w180461913", "w180465548",  
,"w180468951", "w182628812", "w182629658", "w185057057", "w189637127", "w193375648", "w19844790", "w1  
9847245", "w19848882", "w202737485", "w202970121", "w203030595", "w217421358", "w222393302", "w223415  
74", "w22343527", "w22343529", "w22345126", "w225625930", "w22568170", "w22568182", "w22568375", "w225  
68378", "w22568386", "w22568388", "w22568424", "w22568488", "w22568523", "w22588043", "w22588046", "w2  
2604437", "w22673740", "w22674191", "w22674884", "w22675194", "w22698778", "w227400308", "w227400309",  
,"w23025712", "w277863607", "w28353561", "w29619599", "w29619604", "w315383203", "w31662553", "w31662  
878", "w31662879", "w31662880", "w31662964", "w31662965", "w31662968", "w31663124", "w330925134", "w33  
0925135", "w330926195", "w331028474", "w331028475", "w339715316", "w339726718", "w339726719", "w34602  
7918", "w349277858", "w349531415", "w35542048", "w363185344", "w364068225", "w365390727", "w370956707  
", "w38742540", "w38742541", "w40372065", "w40425070", "w40425074", "w4229251", "w443334075", "w443366  
920", "w447751613", "w447751614", "w48225928", "w492147752", "w49835957", "w49835959", "w49835960", "w  
50673023", "w509055318", "w509246741", "w509246744", "w509246745", "w509246746", "w509246747", "w5092  
46753", "w51341167", "w51341168", "w515490602", "w51692324", "w52275569", "w52303054", "w52303060", "w  
525662191", "w525662193", "w52613839", "w52613863", "w526190596", "w530531180", "w530531181", "w53053  
1182", "w531463493", "w531692522", "w531692526", "w531692527", "w536092843", "w536092847", "w55331523  
5", "w587903986", "w587903987", "w587908623", "w587908627", "w587908630", "w587908634", "w59854247", "  
w628788987", "w630445718", "w630446538", "w630455616", "w636627411", "w636627412", "w641622109", "w64  
1622111", "w641622112", "w641622114", "w641622115", "w641622116", "w641622118", "w650553749", "w66200  
6651", "w662006652", "w662006654", "w694476166", "w698066552", "w705274809", "w705274810", "w70527481  
1", "w708338741", "w764629037", "w770613905", "w770613907", "w829119358", "w829119361", "w829119362",  
,"w829119363", "w829119365", "w829119366", "w829180626", "w829180633", "w829252460", "w85552224", "w87  
565425", "w879402866", "w880518011", "w881591456", "w881764531", "w881764532", "w881764537", "w881764  
538", "w881764539", "w881764540", "w881764543", "w881764545", "w881764546", "w881764547", "w881764549  
", "w881764552", "w88646884", "w88646885", "w88646897", "w89578796", "w89685774", "w89685814", "w89685  
825", "w930154230", "w930154231", "w944384994", "w963397962", "w963397963", "w991218943", "w991218944  
", "w991218945", "w991218946", "w991218947"};
```

```
input_file = fopen("france.opl", "r");  
output_file = fopen("output2.txt", "w");  
int find_line = 0;
```

```
if (input_file == NULL)  
{  
    printf("error input");  
    exit(EXIT_FAILURE);  
}  
if (!output_file) {  
    printf("error output");  
    exit(EXIT_FAILURE);  
}
```

```

}

while ((read = getline(&line, &len, input_file)) != -1)
{
    find_line = 0;
    for (int i = 0; i < 256; i++)
    {
        if(StartsWith(line, ids[i]))
        {
            find_line = 1;
            new_line = str_replace(line, " T", " Tcamera=yes,");
            fwrite(new_line, 1, strlen(new_line), output_file);
            break;
            //printf("%s", line);
        }
    }

    if(!find_line)
        fwrite(line, 1, strlen(line), output_file);
}

fclose(input_file);
fclose(output_file);

if (line)
    free(line);
exit(EXIT_SUCCESS);
}

```

# Générer les fichiers brouter

Les script dans le repo fonctionne pas (super :/)

Là une issue ou il y a un script ok. <https://github.com/abrensch/brouter/issues/199>

```

#!/bin/bash
set -e

```

```

# Added
JAVA='/usr/bin/java -Xmx6144m -Xms6144m -Xmn256m'
BROUTER_PROFILES=$(realpath "../../profiles2")
BROUTER_JAR=$(realpath $(ls ../.././brouter-server/build/libs/brouter-*-all.jar))
OSMOSIS_JAR=$(realpath "../../pbfparser/osmosis.jar")
PROTOBUF_JAR=$(realpath "../../pbfparser/protobuf.jar")
PBFPARSER_JAR=$(realpath "../../pbfparser/pbfparser.jar")
PLANET_FILE=${PLANET_FILE:-$(realpath "./france-latest.osm.pbf")} # (!) expects PLANET_FILE to
be set OR 'planet-latest.osm.pbf'
SRTM_PATH=/home/user/workspace/brouter_original/misc/scripts/mapcreation/srtm

rm -rf planet-old.osm.pbf
rm -rf planet-new.osm.pbf
touch mapsnapshpttime.txt

rm -rf tmp

mkdir tmp
cd tmp
mkdir nodetiles
mkdir waytiles
mkdir waytiles55
mkdir nodes55

$JAVA -cp ${OSMOSIS_JAR}:${PROTOBUF_JAR}:${PBFPARSER_JAR}:${BROUTER_JAR} \
-Ddeletetmpfiles=true -DuseDenseMaps=true \
bttools.util.StackSampler bttools.mapcreator.OsmFastCutter \
${BROUTER_PROFILES}/lookups.dat nodetiles waytiles nodes55 waytiles55 \
bordertnids.dat relations.dat restrictions.dat \
${BROUTER_PROFILES}/all.brf ${BROUTER_PROFILES}/trekking.brf
${BROUTER_PROFILES}/softaccess.brf \
${PLANET_FILE}

printf "\n\n----- unotes55 ----- \n\n\n"
mkdir unodes55
$JAVA -cp ${BROUTER_JAR} -Ddeletetmpfiles=true -DuseDenseMaps=true bttools.util.StackSampler \
bttools.mapcreator.PosUnifier nodes55 unodes55 bordertnids.dat bordertnodes.dat ${SRTM_PATH}

printf "\n\n----- segments ----- \n\n\n"

```

```
mkdir segments
$JAVA -cp ${BROUTER_JAR} -DuseDenseMaps=true -DskipEncodingCheck=true btools.util.StackSampler
\
[]btools.mapcreator.WayLinker unodes55 waytiles55 bordernodes.dat restrictions.dat
${BROUTER_PROFILES}/lookups.dat \
[]${BROUTER_PROFILES}/all.brf segments rd5

cd ..

rm -rf segments
mv tmp/segments segments
touch -r mapsnapshpttime.txt segments/*.rd5
```

## Profils brouter

C'est pas ouf, mais pour le POC, j'ai réussi à modifier le fichier trekking en mettant:

```
assign camera = camera=yes

assign turncost = if camera then 86000 else if is_ldcr then 0
                  else if junction=roundabout then 0
                  else 90
```

## Pour tester

brouter-web (c'est du statique)

<https://github.com/nrenner/brouter-web>

## Tiles from postgis

```
export DATABASE_URL=postgresql://postgres:password@localhost/postgres
~/bin/pg_tileserv_latest_linux/pg_tileserv
```

---

Révision #16

Créé 2023-02-28 19:38:10 UTC par tjiho

Mis à jour 2023-03-11 13:09:37 UTC par tjiho