



Andrew Philippides <a.philippides@gmail.com>

tiling Google Maps

1 message

Alex Dewar <a.dewar@sussex.ac.uk>

31 January 2013 20:10

To: "Gaffin, Douglas D." <ddgaffin@ou.edu>

Cc: Andy Philippides <andrewop@sussex.ac.uk>, Paul Graham <paulgr@sussex.ac.uk>

Hi Doug,

I've finally got a working version of some code to tile Google Maps. I haven't put "help" in the files, so I'll talk you through it briefly. You'll need the circular statistics toolbox for Matlab. If you don't have it, you can get it here: <http://www.mathworks.co.uk/matlabcentral/fileexchange/10676>

You need all the folders in the zip file in your "path".

There's code for grabbing images between certain GPS coordinates or over a certain area specified in metres. (All the distances are in metres and I've included functions for finding the distance between GPS coordinates etc.) Images taken from Google Maps are pooled into files in a data folder -- I've called these collections of image files "superimages". You specify the GPS coordinates of the top left corner of the superimage by default, but you can use negative distances if you want to specify the southernmost point of the image, say.

Firstly, to make a superimage:

```
brighton_lat = 50.842941; % latitude and longitude of brighton
```

```
brighton_long = 0.1313120;
```

```
[lat,long] = gps_adddist(brighton_lat,brighton_long,5000,-5000); % find a point 5km to the north and 5km to the west of brighton
```

```
gm_makesuperim(lat,long,'distance',[10000 10000],'zoom',15,'grayscale',false,'label','brighton'); % get a 10km x 10km superimage of brighton
```

This should leave you with some data files starting with "brighton_" in a folder called "data".

To get a superimage between certain GPS coordinates, you can do:

```
gm_makesuperim(50,0,'endcoords',[51 1],'label','brighton');
```

To get at the data you've just saved, you need to make a SuperImage object:

```
superim = SuperImage('brighton');
```

Then you can do things like:

```
res = size(superim); % get total size of superimage in pixels
```

```
im = superim(1:10,:); % get top 10 rows of pixels from superimage
```

I've included functions for converting between GPS and pixel coordinates, because it's a pain to work out:

```
[lat,long] = superim.pix2gps(100,100); % find latitude and longitude for the pixel at 100,100
```

```
[i,j] = superim.gps2pix(50.9,0.2); % find which pixel is nearest to GPS position 50.9,0.2
```

```
dist = superim.pix_finddist(1,1,100,100); % find distance in metres between 1,1 and 100,100
```

Lastly, there's a function for getting (circular) views from the perspective of an agent, centred on the specified pixel coordinates and at a specified rotation:

```
im = superim.getumwelt(100,100,45); % get the view at 100,100 at a rotation of 45 degrees
```

(Note that the image is actually rotated -45 degrees, because that's how the world would look to an agent at 45 degrees.)

At the moment, the default size of these "umwelts" is 101x101 pixels, but you can change it like so:

```
superim.UmweltSize = [201 201];
```

01/02/2013

Gmail - tiling Google Maps

Hope this is clear. I'm happy to give you a demo in person, but thought that either way it's probably good to have a reference.

Cheers,
Alex



google maps tiler.zip

12K