

OpenTrail: augmented reality for walkers

Nick Whitelegg
Maritime and Technology Faculty
Southampton Solent University

Alternative navigation

- ★ The traditional map has been the usual way to navigate in the countryside
- ★ However, with new web and mobile technology, need it be the only way?
- ★ What about augmented reality?

Background: OpenTrailView (OTV)

- ★ Presented at SOTM Girona 2010
- ★ Google StreetView-like application for footpaths
- ★ Users take a series of photo shots at key locations, stitch them into a panorama, and upload
- ★ GPS track with timestamps used to place the panoramas
- ★ See opentrailview.org

OTV: barrier to entry

- ★ Take-up of user contributions to OTV a little slow
- ★ Likely to be the barrier to entry: taking panoramic photos, stitching them together and uploading to the site requires significant effort
- ★ OpenTrailView remains active, but in view of this issue, other areas are being explored...

Augmented reality

- ★ Real-world, as seen through the phone's camera feed, is "augmented" with geographical data
- ★ Common use: overlaying POIs on camera feed for navigation or "live city guide" e.g. Wikitude [1]
- ★ Potential use for countryside navigation

OpenTrail

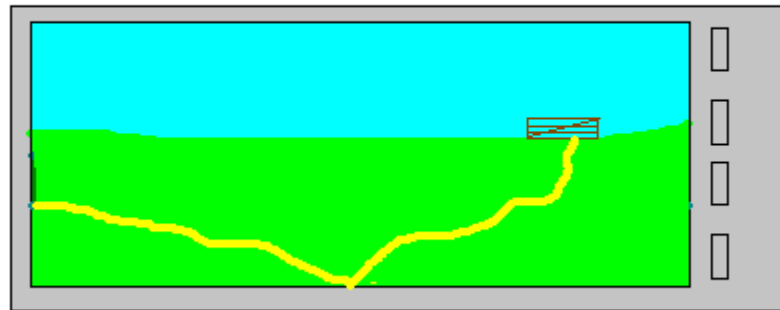
- ★ Android app for countryside navigation
- ★ Shows Freemap (free-map.org.uk) maps by default, or out-of-copyright Ordnance Survey maps from ooc.openstreetmap.org
- ★ Augmented-reality view as prototype

OpenTrail and AR: how does it work?

★ Components:

- ★ Camera feed from phone
- ★ OpenStreetMap data overlaid on feed
- ★ Freely available height data (e.g. Ordnance Survey Land-Form PANORAMA) [2] used to adjust OSM data for height

Intended AR view of OpenTrail



OpenTrail and OSM data

- ★ Local OSM server setup on free-map.org.uk
- ★ Delivers data from PostGIS within a given bounding box in a format optimised for rendering (ways contain coordinates)

★ e.g.

```
<rdata>
```

```
<projection>epsg:3785</projection>
```

```
<way>
```

```
<point x="-85640.13" y="6629853.55"/>
```

```
<point x="-85689.09" y="6629937.75"/>
```

```
<point x="-85722.33" y="6629999.99"/>
```

```
<tag k="highway" v="path" /> ...
```

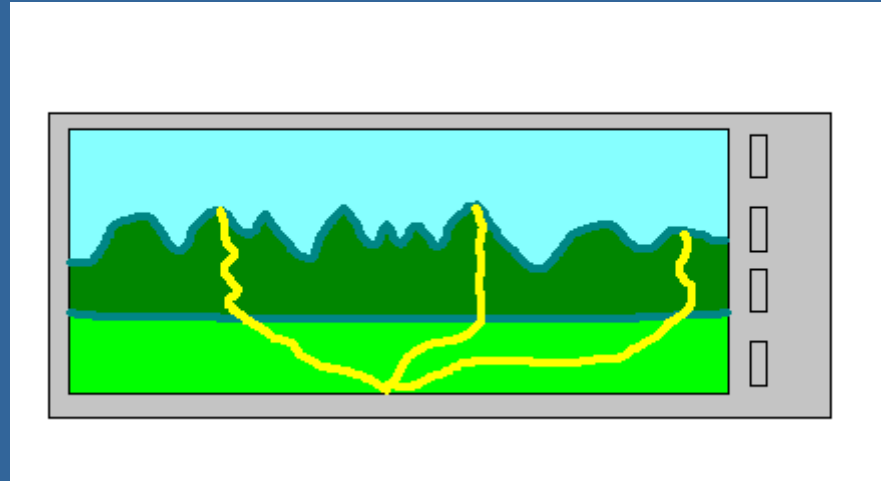
Delivering OSM data to the device

- ★ World divided into approx 5km x 5km tiles (5000x5000 “metres” in selected projection)
- ★ When walking and you enter a new tile, OSM data is downloaded and cached on the phone
- ★ This data will be used for overlaying on the camera feed
- ★ Cache used next time you're in the area

Height data

- ★ Ideally, height data needs to be accounted for
- ★ If there's a hill in the way, it would be desirable to overlay the OSM data on the slope of the hill, rather than “burrowing into the hill”...

OpenTrail and height data



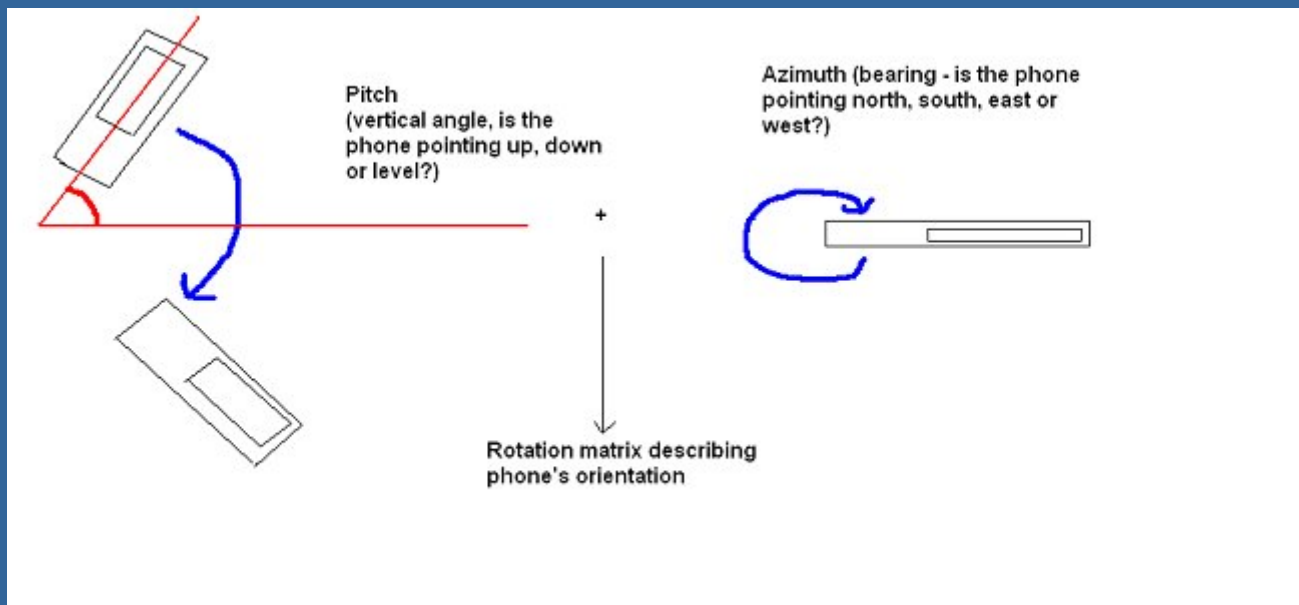
Height data

- ★ Facility implemented but not yet “switched on”
- ★ As part of the OpenData package, the Ordnance Survey provide the Land-Form PANORAMA dataset [2]
- ★ Globally, NASA SRTM[3] or other national datasets could be used
- ★ Gridded heights at 10m intervals
- ★ To obtain the height of an arbitrary point on earth (e.g. within the course of a footpath), bilinear interpolation is used

How is the AR done?

- ★ We need to know our **current position** and the **device orientation**
- ★ Obtain GPS position using the Location APIs[4]
- ★ Use the Android Sensor API [5] to obtain the orientation of the device (bearing and pitch) expressed as a rotation matrix
- ★ This matrix represents the rotation from the phone in standard orientation to the actual orientation

Phone orientation



Rendering the data

- ★ OpenGL [6] is used to render the map data on the device
- ★ OpenGL defines a "modelview matrix" [7] which describes the relation between the 3D world and our current view of it
- ★ The matrix returned from the sensor API can thus be used as the modelview matrix

Current issues

In tests the following appear to be the main issues:

- ★ Inaccuracies in OSM data
- ★ Instability of compass
 - ★ Exponential smoothing [8] [9] mitigates this
- ★ Inaccurate horizontal field-of-view angle

Inaccurate horizontal field-of-view angle

- ★ Android 2.2+ has an API call for obtaining HFOV [10]
- ★ HFOV needs to be accurate to equate the OpenGL perspective with the real world using `GLU.gluPerspective()`
- ★ However Android 2.1 (the test environment) does not :-(
- ★ Worse, no published specs for HFOV, focal length, etc, by device manufacturer
- ★ Manual calibration will be necessary

Issues with data overlay

- ★ Currently, a view showing the camera feed is overlaid on the OpenGL view (yes, that way round! [11])
- ★ However a Google employee has stated that this is not guaranteed to work [12]
- ★ Two possible solutions:
 - ★ Use a single SurfaceView to render camera feed and OpenGL data (tricky, requires low level APIs)
 - ★ Render the camera feed as a texture on the GLSurfaceView (tricky again and slow as camera feed not in RGB format, needs converting)
- ★ Hope to have one of these done for Denver! :-)

Downloading OpenTrail

- ★ Considered “researchware” at the moment until issues resolved, so available as source only
- ★ Source (GPL) in SVN at
<http://www.free-map.org.uk/svn/freemap/android/opentrail>
- ★ APK for basic version (map only, no AR, UK-orientated) at
<http://www.free-map.org.uk/downloads/android/opentrail.apk>

Demo

- ★ Data for Vienna pre-downloaded to the test device
- ★ If there is interest I can arrange a demo in the Resselpark
- ★ Disclaimer: untested outside UK but “ought to” work :-)

References

- [1] Wikitude (n.d) "Wikitude" (online) Available: <http://www.wikitude.com/en/>, last accessed 6/6/11
- [2] Ordnance Survey (2011) "Land-Form PANORAMA - small-scale height data of Great Britain", (online) Available: <http://www.ordnancesurvey.co.uk/oswebsite/products/land-form-panorama/index.html>, last accessed 6/6/11
- [3] NASA (2009) "Shuttle Radar Topography Mission", (online) Available: <http://www2.jpl.nasa.gov/srtm/>, last accessed 6/6/11
- [4] Google (2011) "Location and Maps" (online), Available: <http://developer.android.com/guide/topics/location/index.html>, last accessed 6/6/11
- [5] Google (2011) "SensorManager" (online), Available: <http://developer.android.com/reference/android/hardware/SensorManager.html>, last accessed 6/6/11
- [6] Khronos Group (1997) "OpenGL - The Industry Standard for High Performance Graphics" (online), Available: <http://www.opengl.org/>, last accessed 6/6/11
- [7] Neider, J., Davis, T. and Woo, M. (1994) "OpenGL Programming Guide" Chapter 3, (online) available: <http://fly.cc.fer.hr/~unreal/theredbook/chapter03.html>, last accessed 6/6/11
- [8] Haseman, C. (2009) "Augmented Reality on Android: Using GPS and the Accelerometer" (online), Available: <http://www.devx.com/wireless/Article/43005/0/page/2>, last accessed 10/7/11
- [9] Andreasson, M. (2010) "Technical: Orientation sensing in Photos Around" (online), Available: <http://photosaround.mandreasson.com/2010/05/technical-orientation-sensing-in-photos.html>, last accessed 10/7/11
- [10] Google (2011) "Camera.Parameters" (online), Available: <http://developer.android.com/reference/android/hardware/Camera.Parameters.html>, last accessed 10/7/11
- [11] Henze, N. (2009) "Things about stuff: Camera image as an OpenGL texture on top of the native camera viewfinder" (online), Available: <http://nhenze.net/?p=172>, last accessed 10/7/11
- [12] Hackborn, D. (2009) "Two surface views in a FrameLayout", thread in android-developers newsgroup (online), Available: http://groups.google.com/group/android-developers/browse_thread/thread/08e7df2e96a7973d, last accessed 10/7/11