

CITY UNIVERSITY

Use of a GIS for Production and Maintenance of Street Orienteering Maps:
Can a GIS and Spatially Aware Data add Value to Orienteering?

Oliver G O'Brien

September 2008

Submitted in partial fulfilment of the requirements for
the degree of MSc in Geographic Information Systems

Abstract

Street-O maps, used for informal orienteering races in urban areas, often have a minimalistic style, consisting mainly of black lines representing roads and paths. A new way of producing them is proposed – using a filtered subset of data from OpenStreetMap as a base map, Quantum GIS to manage data from several sources and apply the cartographical styling, and a centralised PostGIS geo-database to store event locations and other orienteering-specific data. The orienteering data is edited in Quantum GIS and the base map can be edited at source if needed. Quantum GIS's map composer is also used to produce the final map, negating the need to transfer the data to a traditional cartography package. Derived contours and examples of points of interest are added as an option, from external sources. The design is implemented in two areas in inner city London, and the base map and orienteering data is also combined to produce ideal routes between points, which are overlaid in Quantum GIS. The whole process is achieved using free, cross-platform software, allowing for straightforward adoption by orienteering clubs.

Table of Contents

Abstract	2
Table of Contents	3
Acknowledgements	6
Terms and Acronyms	7
Organisations.....	7
Cartographical Standards	7
Data Formats	7
Terms	8
1 Introduction	10
1.1 The Sport of Orienteering.....	10
1.2 Orienteering Mapping	10
1.3 GIS and Orienteering Maps	11
1.4 Street-O Mapping	11
1.5 GIS and Street-O Maps	12
1.6 Scope.....	12
1.7 Structure of Dissertation.....	13
2 Literature Review	14
2.1 GIS and Orienteering Maps	14
2.2 Converting to GIS	15
3 Requirements Analysis	16
3.1 Questionnaire on Street-O Map Production	16
3.2 Cartography for Street-O Maps	21
3.3 Considerations in Choosing the GIS Application	25
3.4 Review of GIS Applications.....	26
3.5 Considerations in Choosing the Geo-Database.....	31
3.6 Review of Geo-Databases.....	32
3.7 Considerations in Choosing the Base Map Source	32
3.8 Review of Base Map Sources	33
3.9 Visual Comparison of OpenStreetMap and Existing OCAD Data.....	38

3.10	Considerations in Producing the Map	39
3.11	Review of Techniques to Produce the Map	40
4	Design	41
4.1	The Base Map	41
4.2	The GIS	45
4.3	Managing Orienteering-Specific Data	45
4.4	Producing a Map for Orienteering	47
5	Implementation	50
5.1	Setting up the Software	50
5.2	Introduction to the Map Case Studies	50
5.3	Case Study 1: Putney	50
5.4	Case Study 2: Islington	59
5.5	Using the Data for Optimum Route Analysis	63
5.6	Case Study 3: Routing Between Pairs of Controls	64
5.7	Examples of Solving the TSP	68
6	Evaluation	70
6.1	Shortcomings of the Process	70
6.2	Benefits of the Process	72
6.3	Discussion	73
7	Conclusion	74
7.1	Review of Project Methodology	74
7.2	Recommendations	74
7.3	Addressing the Research Question	75
7.4	Future Research Possibilities	75
	References	77
	Appendices	81
A.	Survey of Techniques Used by Street-O Mappers	81
B.	Customised Version of osm2gml.py to Create Base Map Layers	83
C.	Patch Applied to QGIS to Correct Capping	86
D.	SQL Schema for the Orienteering Tables in PostgreSQL	87
E.	Python Script used to Geo-code the Blue Plaque Addresses	89
F.	Original Putney Street-O Map	90
G.	Putney Street-O Map	91
H.	Islington Street-O Map	92

I. Code Fragments for Routing Street-O Controls	93
J. Dissertation Proposal	95
K. Reflection	101

Acknowledgements

The author would like to thank Dr Jason Dykes, of City University's Information Science department, for his assistance in supervising and providing critical feedback on this dissertation, and Wally and Greg Cavill for detailing and supplying their ShortO application.

The author would also like to thank the following people for completing a questionnaire used in the preparation of this dissertation. Those with an asterisk (*) also supplied map vector files for the project.

- United Kingdom
 - Bristol Orienteering Klub (BOK)
 - Philip Fawkner-Corbett
 - London Orienteering Klubb (LOK)
 - David Rosen*
 - Manchester and District Orienteering Club (MDOC)
 - John Britton
 - Saxons Orienteering Club (SAX)
 - David Kingdon, Nick Barrable
 - South-East Lancashire Orienteering Club (SELOC)
 - Paul Turner
 - South London Orienteers and Wayfarers (SLOW)
 - Ian Ditchfield, Don McKerrow*, Dorte Torpe Hansen*, Mike Garvin*, Steve Allen*, Charles Bromley-Gardner*, David Harrison*, Simon Evans*, Paul Nixon*
 - South Ribble Orienteering Club (SROC)
 - Glenys Ferguson, Karen Nash, Dave Hargreaves
- Australia
 - Bayside Kangaroos Orienteering Club (BK)
 - Michael Hubbert
 - Dandenong Ranges Orienteering Club (DR)
 - Wally Cavill*, Joe Leibowitz
 - Nillumbik Emus Orienteering Club (NE)
 - Ian Stirling*, Geoff Hudson

Terms and Acronyms

Organisations

- IOF** **International Orienteering Federation:** The worldwide governing body for orienteering, it defines the rules and standards of the sport.
- OSM** **OpenStreetMap:** An online project to create a free map and spatial dataset of the world, contributed to by volunteers.
- OGC** **Open Geospatial Consortium:** An organisation that defines and promotes standard geospatial file and data formats, allowing for better interoperability.
- OSGeo** **Open Source Geospatial Foundation:** An organisation that promotes and hosts development of a number of geospatial software applications and libraries.

Cartographical Standards

- ISOM** **International Specification for Orienteering Maps:** The standard that defines the cartographic symbols, styles, colours and sizes by which most orienteering maps, including “regular” foot-orienteering maps, are created. The most recent version is known as ISOM 2000.
- ISSOM** **International Specification for Sprint Orienteering Maps:** A derivative of ISOM, this specifies the cartography for maps used by sprint orienteering, a new sub-discipline of foot-orienteering. There have been a number of iterations, the most recent being known as ISSOM 2007.

Data Formats

- OCD** A binary format developed for storing orienteering and other maps created with the OCAD application.
- DXF** **Drawing Interchange Format:** A format developed by AutoDesk to allow easy data transfer between AutoCAD and other programs. It has been adopted as a de-facto standard.
- GML** **Geographic Markup Language:** An XML-based OGC standard that describes geographic features of various types.

Shapefile	An OGC standard, developed by ESRI, for vector layers.
SLD	Styled Layer Descriptor: An XML-based OGC standard that describes the cartographic styling that should be applied to layers of features.
QML	An XML-based file serialisation of styling used in a vector layer in QGIS.
OSM XML	An XML-based format developed by OSM to distribute their data. An OSM XML file, also known as a “planet.osm” file, contains nodes, ways and relations (which link together multiple ways and nodes).
GPX	GPS Exchange Format: An XML-based format that describes waypoints and tracks collected by a GPS receiver. It is the “de-facto XML standard for lightweight interchange of GPS data” (Foster, 2004).
SVG	Scalable Vector Graphics: An XML-based format for describing vector graphics, it allows for very rich styling and description of complex graphical objects.
Terms	
Node	OSM’s name for a point or corner of a polyline or polygon. Nodes can be unnamed, such as in polyline corners, or named, indicating points of interest such as a mini-roundabout.
Polyline	A line made up of one or more straight lines, joined together and ended with nodes.
Way	OSM’s name for a polyline representing a linear feature on the map.
Segment	OSM’s name for a single straight line making up all or part of a way.
Layer	A set of spatial data of one type, representing one or more features, e.g. all railway lines.
Type	Layers can be made up of polygons, polylines or points – the three most common geometrical types.
Level	Describes the real-world order that features appear in. For example, a railway crossing over a road has a higher level than the road. Called “layer” in OSM, and analogous to layers in vector drawing applications such as Adobe Illustrator.

Feature	A generic term for a single spatial object, such as a road or wooded area. A “runnable” feature, in the context of this project, is normally a linear one on which a runner can run along, for example a road, but not a railway line.
Attribute	Information associated with each feature.
Tag	OSM’s name for a feature’s non-spatial attribute.
Key	The category of the tag, e.g. “highway”.
Value	The value associated with the tag, e.g. “secondary”.
Taxonomy	A defined set of tags that categorise all features. For example, the taxonomy of road types might require a road to have a “highway” key in its set of tags, with a value of “motorway”, “major” or “minor”.
Datatype	Describes how a particular attribute for a layer is stored in a database or file. Typical database datatypes are strings, dates, integers and real numbers.
Stroke	The graphics used to display a polyline. The stroke can have colour, width, and style (e.g. a sequence of dashes) and there may be more than one per polyline, although each appears for the whole length of the polyline.
Cap	How a stroke, of finite width, is rendered at the endpoints of the line.
Project	The legend, view and map in a GIS make up the project.
View	The main pane in a GIS, where the layers are displayed together.
Legend	In the context of a GIS, the legend is the list of currently loaded layers, it offers an interface to show or hide them in the view/map, reorder them or change their attributes, while also functioning as a traditional map legend by showing the current style for each layer.
Map	The map is normally a distinct window or mode in a GIS, allowing a map to be assembled for printing, consisting of part of the view, and a number of adornments, such as a customised static legend, scale bar or title.
Style	The cartographical look for each layer. Styling a GIS layer, for direct creation of a map, is a substitute for passing the data to a dedicated graphics or CAD application.

Please also see the club name acronyms in the Acknowledgements section above.

1 Introduction

1.1 The Sport of Orienteering

Orienteering is a running sport, similar to cross-country but where competitors navigate between control points, using a specially produced map and often a compass. There are four disciplines in the sport (IOF, 2008), the main one being Foot Orienteering, which is the one that this dissertation will focus on.

1.2 Orienteering Mapping

The map is a key component of orienteering, and has evolved along with the rest of the sport. In the 1960s, events were held on black-and-white photocopies of public maps, such as Ordnance Survey (OS) 1:25000 maps in the UK and National Research Council 1:20000 maps in Canada (SROC, 2008; Streb, 2008; Kirk, 2004). The maps were often missing detail encountered by the competitors, and control points were sometimes on features that were not on the map (Sibley, 2007).

Today, orienteering maps are highly precise. They are at 1:15000 or larger scale, printed by offset litho or laser printing with up to six spot colours. On the largest scale maps, used for sprint racing, typically any prominent tree, bush or boulder is shown (Tveite, 2006). Within the UK, there are several professional dedicated orienteering mapping companies (Head, 2006), as well as many amateur mappers, typically producing maps of smaller, local areas for their clubs.

Orienteering maps are typically produced from a base map (often supplied by Ordnance Survey in the UK) or a photogrammetric plot. The mapper walks the area, using tracing paper and pens to annotate features, and taking photographs. GPS tracks are starting to be increasingly used, but it is believed, based on personal knowledge, that there is little use of more sophisticated solutions when in the field, such as laptops or PDAs with portable CAD or GIS applications on them.

Each orienteering map is typically self-contained, produced at a single scale and with a specific symbol set. Maps previously were hand-drawn, but today, the cartography is generally carried out using a CAD or vector graphics application. The vast majority of maps are produced using OCAD (Hendricks, 2005), a CAD with a number of orienteering-specific features that produce attractive looking maps, such as aligning broken line segments for path junctions, intelligent placement of tick-marks on lines, and pre-inclusion of the international orienteering symbol standards. Due to the predominance of the application,

printing companies used by orienteering mappers accept OCAD files as their native format. Adobe Illustrator is used by a small number of mappers, as a cross-platform solution (Zentai, 2000; Trussler, 2004).

1.3 GIS and Orienteering Maps

OCAD and Illustrator are not GIS applications – that is, they primarily produce cartography that is not spatially aware and so does not allow sophisticated spatial calculations to be done on it. There are limited GIS operations that can be done – in OCAD, the map’s scale can be set, so that distances can be displayed when drawing courses, with the (appropriate) large-scale assumption of a planar surface. Recent versions of OCAD do have limited GIS capabilities, where DXF and Shapefile files can be imported for tracing (OCAD, 2008a) and DXF files exported, and a base map raster and the drawn map can both be geo-referenced by selecting common features, allowing a good fit (OCAD, 2008b). But fundamentally the data in OCAD intended for a static, printed map. Routing would be difficult, for example, due to nodes not necessarily being specified, feature layering not being formalised, or segment ends not joining up.

GISes aren’t used for orienteering maps, primarily because there is not a need for a GIS to be in the workflow of producing a standalone orienteering map. The existing process is good enough to produce high quality orienteering maps. Leung (2003) writes that OCAD “is more convenient, faster and accurate than working with the normal features of a GIS to produce a simple map.”

The purpose of this dissertation is to examine the benefits that GISes can bring to enhance the process and the sport, complimenting and possibly streamlining the current process.

1.4 Street-O Mapping

Street-O events are informal, social training events held by some clubs, generally in urban environments, particularly on weekday evenings during the winter season when training in more rural environments is less practical. Instead of visiting orienteering markers at the control points, competitors run with a clue-sheet, answering a question at each location, such as “What is the colour of the door at No. 7?” Typically, the event is a score event, where competitors can visit many control points, in any order, in a fixed time. This allows the event to be run with the minimum of overhead, normally by one person (Barrable, 2007; personal experience).

In the UK and Australia at least, Street-O maps are generally largely black-and-white, with less detail that would be expected from an ISSOM orienteering map. Roads are generally

shown as uniformly thick solid black lines, or atlas-style bordered white areas, not indicative of the road's width. Paths are shown as dashed black lines. Control circles are shown in the normal magenta colour, sometimes with a dot in the middle to show which side of the road the feature lies (Nillumbik Emus, 2008; personal experience). Depending on the area, contours, major bridges, rivers and railway-lines may be shown, but equally could be omitted for clarity (Cameron, 2007). Areal features, such as parks, and point features like benches are generally not shown – the map predominately only shows linear features. Street or area names are generally not shown, except sometimes at the edges. From personal experience, the sparseness of the map can look quite intimidating, while at the same time being clear and unambiguous to interpret.

There is some evidence of full-colour, high-specification orienteering maps being used for Street-O events (Ryyppö, 2007; Wright, 2008). Conversely, photocopied OS maps are also occasionally used (Street, 2001). There is no set standard for how a Street-O map looks, and the choice of cartography is normally dictated by existing maps and the time available to draw. However, the line-based maps described above appear to be the most widely used and are, for the purposes of this dissertation, considered to be the “Street-O” style.

1.5 GIS and Street-O Maps

Street-O maps appear to be ideally suited for use and production in a GIS. Their simple, primarily linear cartography, sparse look, varying scales and lack of complex colours and aesthetic tweaks means they could potentially be produced using the data-based concept of a GIS rather than the artwork-based concept of a CAD or a drawing package.

A GIS database containing Street-O mapping data for a large area, would allow maps of individual areas to be produced, or joined together as appropriate for the event, with the scale used changing depending on the area required and printing constraints. High precision is not particularly important, so, for example, contour information could be automatically derived from freely available altitude data, such as from the Shuttle Radar Topography Mission (Allan, 2008), or topological data from OpenStreetMap. A number of other potential uses of GIS in terms of Street-O maps are studied in this dissertation.

1.6 Scope

This project focuses on Street-O orienteering maps, specifically the predominately black-and-white ones with mainly linear features that are used by South London Orienteers and Wayfarers (SLOW) for their winter series. London Orienteering Klub owns two of the maps concerned, but the area covered and style compliment the SLOW ones.

The software that is used in the project is evaluated from the viewpoint of a club-level, non-professional mapper without a budget. This limits the scope of applications evaluated, to free ones.

The implementation in the project is restricted to construction of a prototype, the server components of which is hosted on a computer not accessible through the Internet, that is, further rollout would be needed for the prototype to be adopted as a working system for use by club mappers.

1.7 Structure of Dissertation

Initially, the results of a questionnaire sent to existing Street-O mappers, are analysed. From this, observations and shortcomings of the current process are understood. The existing cartographical styles used are also studied.

A number of groups of technologies are then considered – with a discussion of the most important aspects of each technology group being followed by a short review of a number of examples in the group. These groups include base map sources, desktop GISes, geo-databases and map producers.

From the above, an appropriate technology in each group is chosen, and a process and configuration, linking them together to create a map, is designed.

The design is then implemented in a number of case studies, to demonstrate a practical answer to the research question.

The methodology that was designed and implemented is then evaluated, looking at its successes and failures.

The dissertation concludes by reviewing the work, making a number of recommendations for practical implementation and ideas for future research.

2 Literature Review

2.1 GIS and Orienteering Maps

Leung (2003) has written an MSc dissertation on the same theme – integrating GIS into orienteering – although with more of an emphasis in the potential of spatial intelligence in course planning rather than on maintaining the underlying map, and concentration on raster-based GIS analysis. He identifies weaknesses in current course planning, including such as necessarily over-simplified calculations of course difficulty. The emphasis is on planning a “conventional” orienteering course, that is, one where the runner visits all the controls in order, the winner being the fastest person around the course.

He eliminates GIS as a possibility for making the orienteering maps themselves, as “no GIS can be customised with a sufficient number of mapping symbols” to ensure an equivalent cartographical representation. He highlights the ability for terrain types to be reclassified as “cost surfaces”, representing the difficulty an orienteer would have passing across or along a line feature or through a area, and allowing optimum route calculations, based on least-cost-surface analysis, to be made for proposed courses. There is included a simple mathematical treatment of factors in such a calculation for a leg.

Of note, the GIS applications analysed are raster-based ones, due to conventional orienteering route choices being across areas rather than along linear features as in Street-O. Aerial imagery is used to classify features, and DEM information is used to construct contours on a map, on which the best route is simulated. Viewshed analysis is used to check that adjacent controls are not visible.

Hendricks (2003 & 2005) has written two short papers for ESRI User Conference events. In the first, he describes using a laptop computer, GPS receiver and ESRI software to quickly create “near-standard” orienteering maps – the technology being used for the fieldwork as well as the cartography. Two customisations are made – a custom form in ArcPad that restricts data input to the ISOM feature taxonomy, and a style sheet to give the data the ISOM look-and-feel in ArcGIS. The survey was done by first recording line features, and then on a subsequent visit point features, both using the GPS receiver. Areal features were sketched manually. Features were generalised and displaced where necessary for clarity. The map was exported as a PDF and printed directly, without further processing in a CAD.

In the second paper, he describes the potential integration of a number of non-georeferenced OCAD maps into a multiple purpose GIS database, highlighting that both the

orienteering maps, and the GIS database can gain from adding each other's data. Due to differences in the scope of the schemas from each data set, an intermediate GIS database, with the ISOM schema, is proposed. Fieldwork would be performed using this schema, and data would be transferred to both the general GIS database for data editing and storage, and optionally OCAD for layout and map design. Metadata is mentioned as a weakness of OCAD.

Karlin (2008) created a DEM from the contour lines on an existing orienteering map, and a friction raster from the vector features on it. The rasters were combined and, using a function, theoretical isochrone maps could be created. From these, idealised route choices for a typical orienteering course could be created. These were compared with choices taken by real orienteers. Oversimplifications with "slope friction" were acknowledged.

Halasz (2005), a developer of the Quantum GIS (QGIS) open-source GIS application, mentions "a future goal of using it for orienteering map production", highlighting the potential of the plug-in system as "a way to extend QGIS to do some interesting things."

Fan (2005) mentions development of an "Orienteering Map Assistance Design System Based on GIS". The paper is in Chinese.

2.2 Converting to GIS

Wunderlich (2008) describes a project that moved a large number of geological maps, in Illustrator format, to reside on a geospatial database (geo-database). Key benefits of the transition were being able to add spatial referencing, attribution and other ancillary information. AutoCAD was used as a staging application between Illustrator and ArcMap for the vector features, to create Shapefiles, a standard format for geospatial vector data, and the background raster maps were georeferenced. These were then added to the geo-database. Conversion of the symbols for point features, retaining the vector direction of the point, required care. The process was "less labour intensive than scratch digitising".

3 Requirements Analysis

3.1 Questionnaire on Street-O Map Production

Using the web and personal experience, four active Street-O series were identified – in London, Kent and Lancashire in England; and in Melbourne in Australia. A questionnaire was emailed to people who had drawn the maps for these series, and some other mappers who had drawn some stand-alone Street-O maps.

The purpose was to find out:

- Their experience with GIS applications and their thoughts regarding the use of these for orienteering maps (Q1 & 2).
- The techniques and timescales for producing Street-O maps (Q3 & 4).
- The process of preparing an existing map for a new event (Q5).
- The software used for the cartography (Q6).
- Their thoughts on a number of ideas that GIS might make possible (Q7).

Overall indications from the results were used to understand the current processes, and guide the scope and requirements of the process that this project developed.

The questionnaire can be seen in Appendix A. It was initially sent to 23 people. Some recipients forwarded on the questions to other Street-O mappers in their club. In all, 23 responses were received. Individual respondents are not cited below for reasons of confidentiality due to the techniques some may have used to produce the map.

As the questionnaire did not ask for quantitative answers, the categorisation in the tables below is derived from the free-text answers and so is indicative only.

The names and countries of the survey respondents are listed in the Acknowledgements section above. Due to the small and already targeted sample, a demographic breakdown was not performed.

3.1.1 Experience of GIS (Q1 & 2)

Just two of the respondents had experience of using a GIS application such as ArcGIS. Four others indicated that they were aware of what it was. Respondents with an awareness of GIS generally thought that it would be “overkill” for “informal”, “simple” Street-O mapping. One person mentioned the benefit of being able to overlay data from multiple sources and have change tracking.

Experience of GIS	No of People
No experience	17
Aware but no active experience	4
Experienced	2

3.1.2 Surveying and Drawing the Map (Q3 & 4)

A number of respondents identified three stages:

1. Tracing from a background map – this was generally identified as the most tedious stage of the process.
2. Carrying out fieldwork, noting details while walking, cycling or “slowly driving” around the streets concerned, with pen and paper. This was identified by many as the longest stage.
3. Making the corrections and updates on the map.

Source for Background Map	No of People
Ordnance Survey maps	7
Google Earth or Google Maps	7
Street and road atlases, e.g. “A-Z”	6
Other raster maps from the web, e.g. “Streetmap.co.uk”	5
Water Board maps	1
Large-scale plans	1
GPS traces	1

The numbers above exceed the number of respondents as some people used multiple sources for different maps. It should be noted that Ordnance Survey maps only are available for the British-based maps.

Four people mentioned that they used aerial imagery for crosschecking their fieldwork.

Only one person had derived their street map from their own measurements, using a GPS. One other person had experimented by recording GPS tracks while running over the area – however they had found it was not possible to distinguish between roads and paths using this method (these are normally shown with different symbols on Street-O maps.)

The time taken to produce the map varied widely, from a few hours to a few days.

Time Taken to Produce Map	No of People
Less than 8 hours	1
8-15 hours	5
16-24 hours	5
More than 24 hours	5
Not specified	2

It should be noted that some respondents did not give time estimates for all the stages in producing the map. “A day” is assumed to be eight hours.

Some comments were:

- “The physical survey probably took the longest – in such a complex urban environment simply to use existing maps... or even aerial survey would not have been sufficient to produce a map even good enough for a low-key Street-O event”
- “Field checking takes longest”
- “One has to walk EVERY possible route choice that is shown on the map”
- “Tracing the contour lines is the most tedious part. However for Street-O Park map precision contour lines are not required”
- “The longest part would be ensuring the smaller streets and connecting paths were shown accurately”
- “Longest part is going over OS scan with black lines”
- “Tracing over the initial scan is fairly tedious”
- “The most frustrating part is the initial tracing of all the roads in OCAD”
- “Getting the initial map to the right scale was a bit of a challenge”
- “Adding the legend is the most frustrating part!”

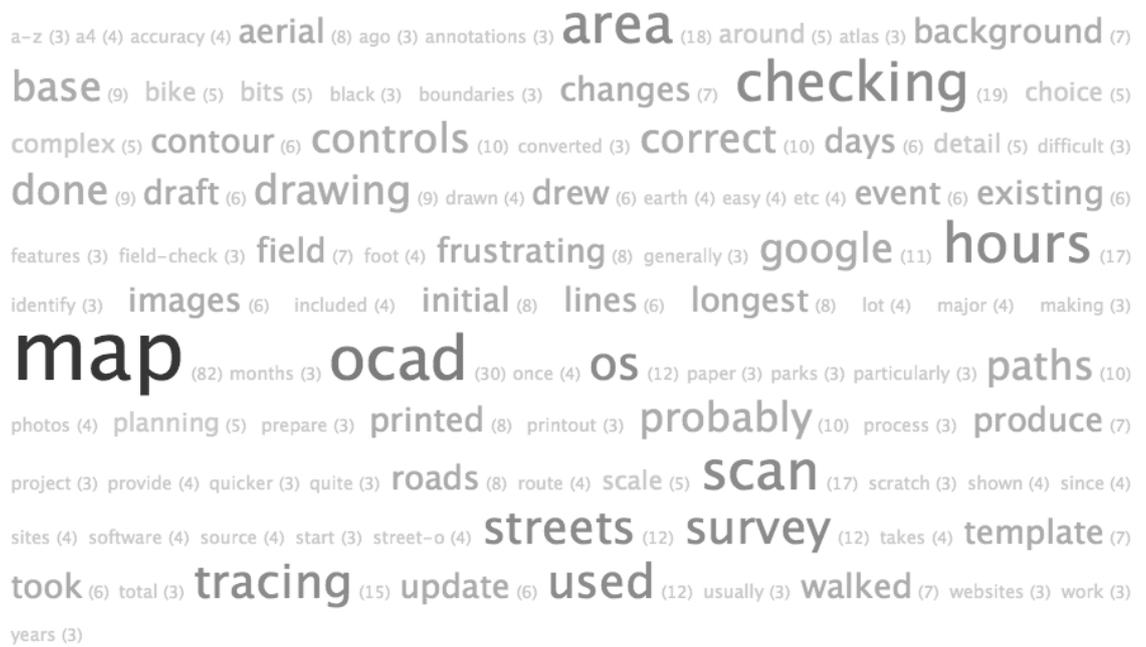


Figure 3-1: Tag cloud of the responses to Q3 & 4, generated at tagcrowd.com

3.1.3 Preparing for an Event (Q5)

In general, the course-setters, who are not always the original mappers, first picked the rough area on the map and then went out and find a suitable control site near the area, noting down map corrections near the control sites that they see, rather than noting down controls at the point of creating the map. Quite a few of the mappers said they would cover the whole area again to check for map corrections.

One person made use of the blue plaque website (English Heritage, 2008) to find interesting control sites.

3.1.4 Software Used (Q6)

All but two of the respondents used the OCAD application to produce their maps. One person used Microsoft PowerPoint for their current Street-O maps, and one hand-drew their most recent map. Justifications for OCAD were that “it makes it easier to share the map with other planners” and “an excellent package... there’s nothing easier.” A couple of people had used other software previously – CorelDraw and Adobe Illustrator – before switching to OCAD for later maps.

Software Currently Used for Street-O Mapping	No of People
OCAD	21
Microsoft PowerPoint	1
Hand drawn	1

3.1.5 Ideas Enabled by GIS for Street-O Maps (Q7)

The first idea put to respondents was an automated “optimum route” calculation for score Street-O events, to perhaps be revealed to competitors after the race.

Most respondents thought that this was an interesting idea – it “provides the interest that is missing without splits”. Of note, a couple of mappers for the Melbourne series mentioned that they already have such a system, using “Travelling Salesman Algorithms to determine the maximum score for each distance” and a “program based on a ‘Directed Graph’ approach, to calculate best routes, for visiting 6 to 20 controls.” Measurement and creation of the distance matrix is “a time consuming process”.

As one of GIS’s potential strengths would be the spatial storage of map meta-data in discrete layers, a couple of ideas relating to such storage were mooted.

One was the storing of potential control sites for future races. Several people mentioned that they already do this in OCAD, storing the sites as objects that are hidden for printing. One person uses “different control shapes to provide indication of control type.” Another believed “an automatic link... between map and clue sheet... would remove a potential source of error.”

The other was the storage of landowner contact information and survey date information, as areal objects in the GIS. Some thought that this was unnecessary for Street-O events. One mentioned that there is already a database of such information for the Melbourne series – they “tried storing such information in maps, but it rapidly falls out-of-date, accuracy questionable”.

Finally, an advanced concept where features on the map would change depending on when they were printed, such as park paths not showing for night-time winter events but showing for day-time summer events, was suggested. Responses were mixed on this – a few thought this was “too clever” and “overkill”. A couple saw more of a potential for “real” orienteering maps, where, for example, summer nettle growth could be shown automatically. For the Melbourne series, maps are themselves classed as winter or summer maps.

3.1.6 Other Information from the Questionnaire

One interesting comment from a respondent – “It would be useful to have an off-the-shelf package to produce or maintain Street-O maps of urban areas – however I cannot see that any system would do away with the thorough walking of the area once the map is

produced which is the most time consuming part of the operation”. Another noted that “It’s all generally done in a very haphazard way”.

3.2 Cartography for Street-O Maps

3.2.1 British Street-O Maps

Maps used for series in the London and Lancaster areas were examined.



Figure 3-2: Extracts from three SLOW maps recently used for Street-O events in London. **Top Left:** Surbiton, drawn by Simon Evans. **Bottom Left:** Wimbledon, drawn by Don McKerrow. **Right:** Lambeth, drawn by Charles Bromley-Gardner.

There are no written instructions for how to produce the Street-O maps for the SLOW series, however the cartography is similar, as new mappers would be familiar with the style of the existing maps. For example, the legends look similar, and paths are consistently represented with dashed lines. The Lambeth map in Figure 3-2 in fact “grew” east across the river from another mapper’s Pimlico map.

There are some significant differences in the three sample maps in Figure 3-2:

- The Wimbledon map uses arrows pointing down-slope to show hills. The cartographer indicated that manually tracing the contours would be too tricky.
- The Lambeth map does not show some of the railway lines. The cartographer mentioned this was for clarity.

- Major roads are drawn with the same stroke as minor roads for Wimbledon, thicker lines or two parallel lines for Lambeth, and thick brown areal lines for Surbiton.

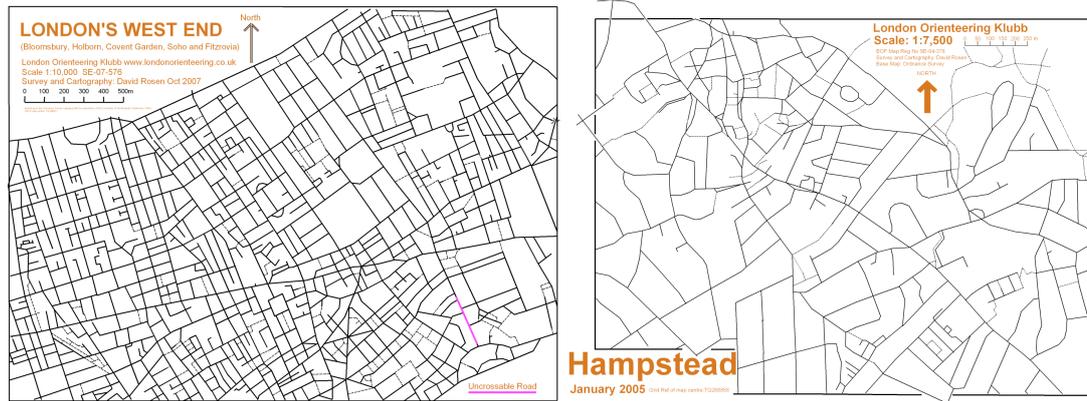


Figure 3-3: Two Street-O maps drawn by David Rosen for LOK events.

David Rosen’s maps for LOK in Figure 3-3 are even more minimalistic. On the left - the number of streets in London’s commercial centre (Oxford Street runs horizontally through the middle) means this map becomes an intimidating mass of black lines, if you lose your concentration for the race – very much a deliberate style to test people’s skills. The Hampstead map on the right has the same simple style, the street network more chaotic but less dense. Hampstead is a hilly suburb, and the lack of contours on the map could potentially catch people out.

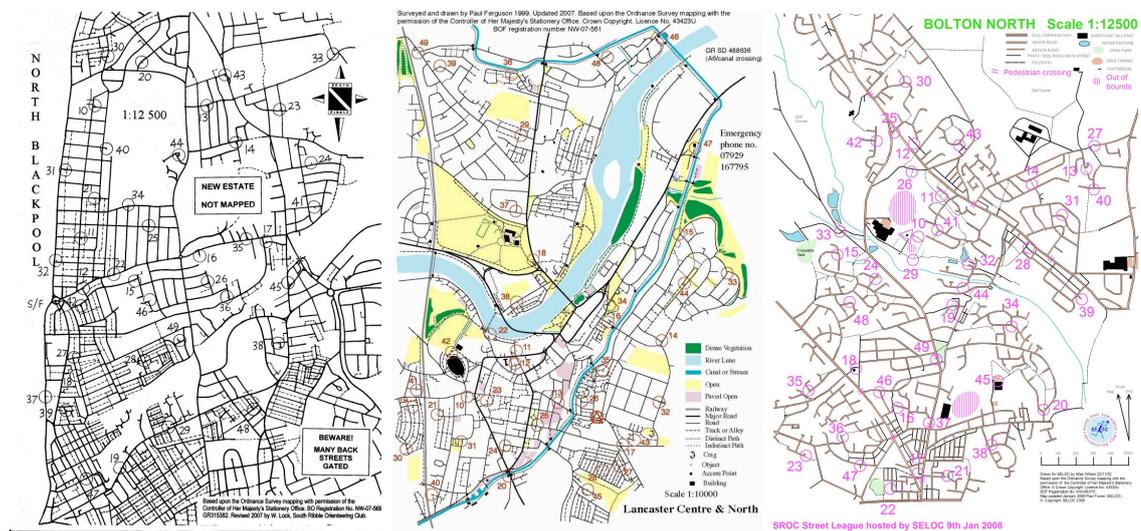


Figure 3-4: Street-O maps created for SROC and SELOC events. **Left:** Blackpool, drawn by Bill Lock. **Middle:** Lancaster, drawn by Paul Ferguson. **Right:** Bolton, drawn by Allan Wilson.

The Lancashire maps vary more in design, while retaining the basic style of un-shaded residential areas, linear road representation, and dashed paths. Various text captions are included on the maps – captions are not normally included on ISOM/ISSOM maps. They indicate major features such as hospitals, hazards, and unmapped areas.

3.2.2 Melbourne Series Maps

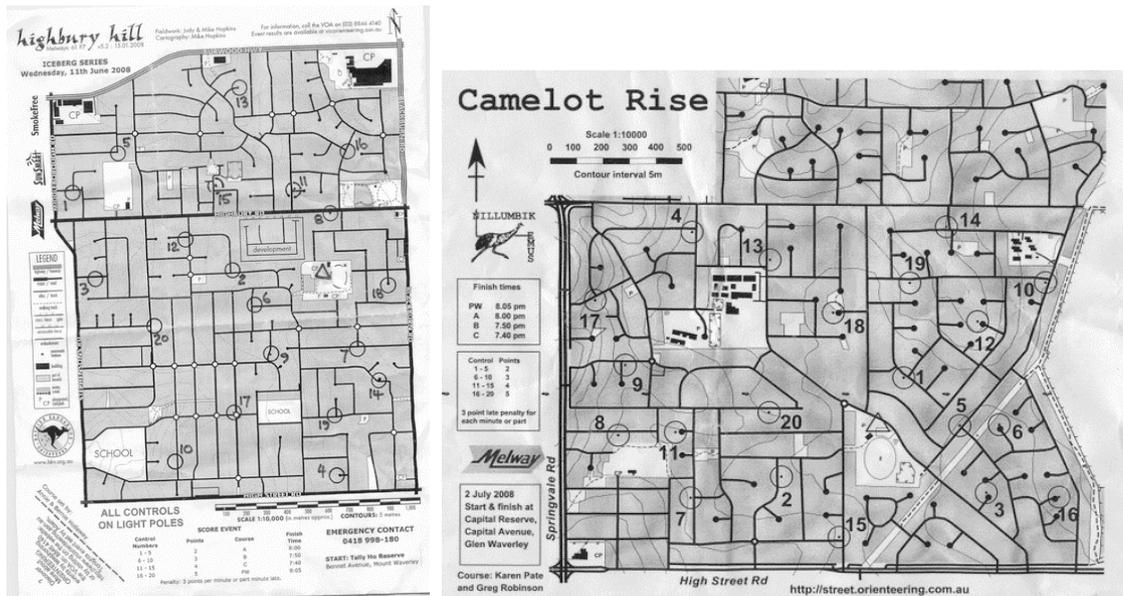


Figure 3-5: Two sample Street-O maps from Melbourne in Australia, produced for the Bayside Kangaroos and Nillumbik Emus orienteering clubs. **Left:** Highbury Hill, drawn by Mike Hopkins. **Right:** Camelot Rise.

A number of these maps are downloadable from the series website, which includes a style guide document, helping ensure a consistent appearance (Nillumbik Emus, 2008). The main difference, compared with the British style, is that the out-of-bounds residential areas are shown shaded grey. The maps also contain many adornments, including the names of boundary roads and sponsor logos and often an emergency phone number, legend and a scale bar. The maps are black and white photocopies. The scales vary from 1:8000 to 1:12500.

3.2.3 Other Styles

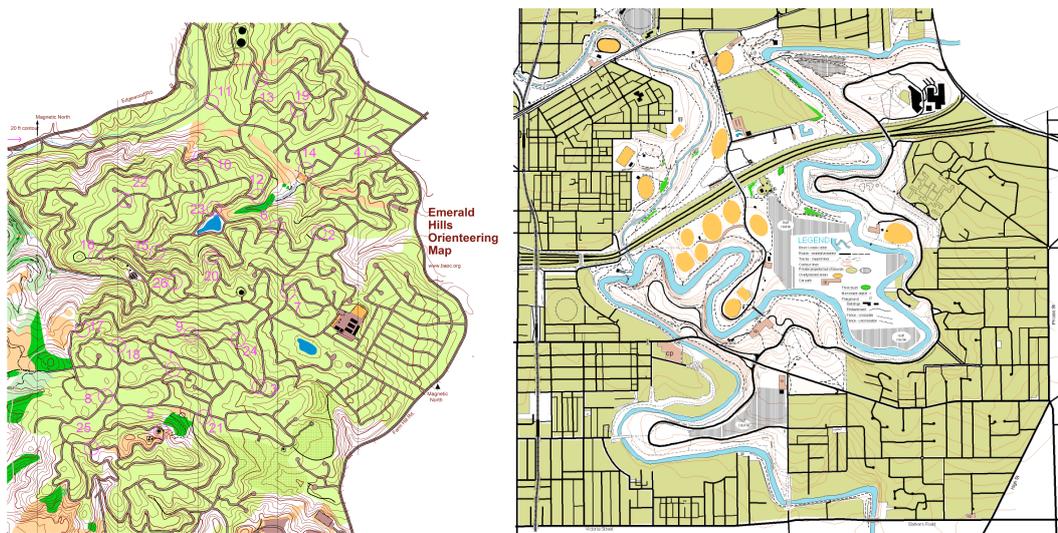


Figure 3-6: Left: A US Street-O map produced by BAOC. Right: A Melbourne Street-O map produced by NE. Both maps combine a complex street network with traditional wooded orienteering terrain, using ISOM symbology and colours, including olive-green to represent out-of-bounds areas, rather than white in the British maps.

These full colour maps are a hybrid between the minimalistic style used for British Street-O events, and a “proper” ISOM orienteering map. Most of the map has the Street-O “network” style, but ISOM symbology and colours are used throughout, showing residential and wooded areas on the map, water features and detailed contours.

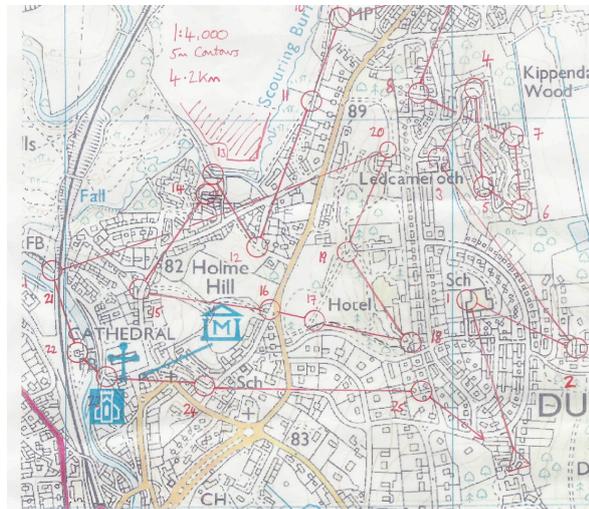


Figure 3-7: Extract from a Street-O course on an Ordnance Survey map.
Base map © Crown copyright Ordnance Survey. All rights reserved.

Not all Street-O events are run on specially produced maps – this linear street course in Dunblane, Scotland, was run on a colour copy of a 1:25000 Ordnance Survey map (Dean, 2007).

This dissertation will concentrate on Street-O maps with SLOW’s design, which is a good balance between simplicity, clarity and realism.

3.2.4 Cartographical Comparison of Some SLOW Street-O Maps

As noted above, SLOW maps do not have a formal style guide, but mappers generally use an earlier Street-O map as a guide when producing their own map. Mappers also generally use the ISOM symbology that they would be familiar with from regular orienteering maps.

Map	Scale	Major Roads	Minor Roads	Paths	Bridges	Rail	Water	Contours	Adornments	Notes
Southbank & The City	1:10000				Incomplete, for major roads only			None	L, N, S	1 (Magenta), 2
Battersea & Chelsea	1:10000				Incomplete, for railways only			None	L, N, S	3
Dorking	1:10000				Not shown except by layering			Curved brown lines, with indexes		4, 5 (Yellow)
Fulham	1:11000				Shown			None	N, S	5 (Green), 6
Guildford	1:15000				Shown as black bars			Green up-arrows	L, N, S	2, 5 (Yellow)

Kensington	1:13333				n/a	n/a	n/a	None		
Lambeth	1:10000				Shown	Some not shown		None	L, N, S	
Pimlico	1:10000				Shown		Not shown	None	L, N	
Putney	1:10000				Shown			None	S	7
Raynes Park	1:10500				Shown		n/a	Red up-arrows		1 (Black)
Richmond	1:11500				Shown, some as tunnels		Not shown	None		5 (Yellow)
Surbiton	1:12500				Shown			None	N, S	
Wimbledon	1:10000				Shown			Magenta up-arrows	L, N	1 (Magenta)

Table 3-1: Symbology used on a selection of Street-O maps produced for SLOW events.

Notes on Table 3-1:

For bridges, the style shown is the familiar “parapet” style, unless otherwise indicated.

For adornments, L = Legend, N = North Arrow, S = Scale Bar.

1. Some areas on the map are shown with out-of-bounds hashing.
2. Some major buildings, such as train stations, are shown as black areal features.
3. Roads in Battersea Park are shown as thicker dashed lines.
4. Forests and paved areas are shown with green and brown shading.
5. Grassland is shown with yellow shading.
6. Cemeteries are marked with crosses.
7. Caption “The Thames” is used to indicate the location of the river.

From Table 3-1, it can be seen that:

- The most common scale is 1:10000.
- There is often a distinction between major and minor roads.
- Railway lines are almost always, included, using with the same symbology.
- Bridges are generally shown, but not always consistency.
- Water features are shown as areal or linear depending on the context.
- Contours are generally not included.
- Map adornments are not always included.

3.3 Considerations in Choosing the GIS Application

3.3.1 Ease of Use

Key to the choice of a GIS is its ease of use. Most orienteering mappers are not professional cartographers or GIS users, and will not be familiar with GISes, as the questionnaire responses showed. In addition, GIS applications are notorious for having a

steep learning curve and being unintuitive and over-featured. This can be a particular problem with open-source applications, where addition of new features can sometimes takes precedence over creating a stable program with a logical, consistent UI. There does however seem to be an opinion in the community (Haklay, 2002 & 2008a; Berlin 2008; Tiits 2003) that GISes in general are hard to use. For non-professional mappers, such as the street-O mappers, whom the survey showed are overwhelmingly not GIS-experienced, it is essential that any process using a GIS is straightforward.

3.3.2 Cost

Orienteering is an amateur sport in the UK, and most small maps are produced by club-level amateurs, with only a few professional orienteering mappers producing maps for major competitions. It is important for the GIS application to be free or low-cost to use, at least for non-commercial purposes, so that it can be used by anyone who wants to, without hindrance.

3.3.3 Platform Independence

Ideally, the GIS will be able to be used on as many operating systems and devices as possible, to allow as many people as possible to utilise their existing hardware to produce and maintain the orienteering map data. These might include: Mac OS X, Windows XP/Vista and Linux computers, and mobile devices, such as Symbian and PocketPC based smart-phones and the Apple iPhone.

3.4 Review of GIS Applications

The shortlist of applications reviewed here was derived from OSGeo's website, and from personal experience. This is not intended to be an in-depth comparison of the applications, rather a summary of their usefulness for this project.

3.4.1 GRASS (6.3.0)

This is the most established and complete open-source GIS – it has been in development since 1982, and is more of an entire platform than just a GUI. It is particularly strong for spatial analysis, delivered in over 300 modules (GRASS, 2008), including a travelling salesman problem (TSP) solver. It also imports DXF files, the vector format that the existing Street-O maps in OCAD can be exported to. It can connect to the main open-source geo-databases and many other systems.

Its multi-windowed interface, huge number of menu items and command line make it very powerful, but also intimidating for the new user, particularly for someone who is not used to how GISes work. It is very much orientated towards having giving the user as much

choice as possible, rather than taking a streamlined approach. The four tabs, eleven checkboxes, three fields and five buttons in the DXF import GUI are a case in point. From personal experience, installing and running the application is not a smooth process.

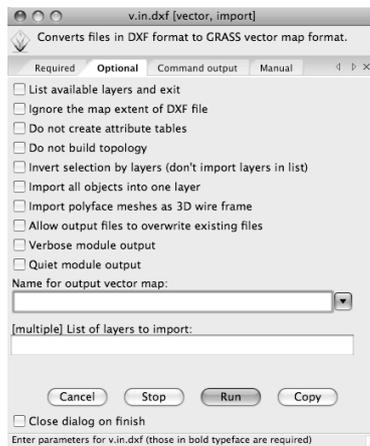


Figure 3-8: Part of the DXF Import dialog box.

3.4.2 Quantum GIS (1.0.0pre)

Quantum GIS (QGIS) is designed as a “user friendly” application and has good integration with the PostGIS geo-database. It has few built-in features for geospatial analysis, but can interface with GRASS’s collection of analytical tools, via its plug-in system. It cannot directly import DXF files, but very recently added a plug-in that can convert DXF files to shapefile format. It can import GML files as read-only, and Shapefiles – it can convert the former to the latter.

Importantly, it has a well-designed and attractive user interface.

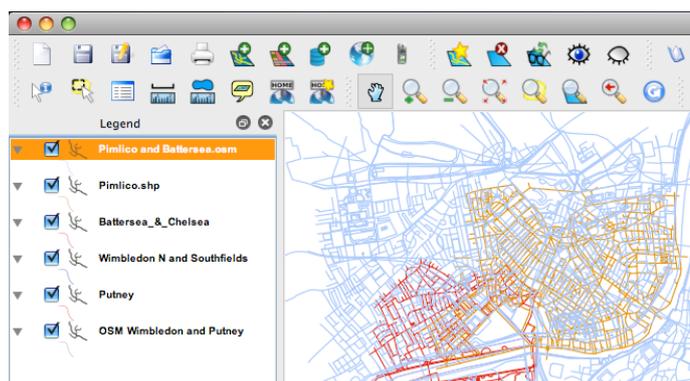


Figure 3-9: Part of the main window in Quantum GIS.

QGIS has very good integration with PostGIS – it was built “ground up” to interface well with it. However, it does not currently directly integrate with MySQL or Oracle Spatial.

It is easy to install and intuitive to use, and is cross-platform. It is in active development with version 1.0 due to be released later in late 2008 – a pre-release build was tested.

QGIS does not offer many built-in geoprocessing capabilities, such as intersection detection. It has an interface to GRASS – geoprocessing can be performed through this. A plug-in architecture allows additional functionality, written in Python or C++, to be easily added.

Some excellent papers have been written introducing QGIS, such as Sherman (2007).

3.4.3 uDIG (1.1rc14)

Like QGIS, uDIG is easy to install and load, and has an attractive, clean looking interface, although it is not quite as intuitive as QGIS to use. It opens Shapefiles but not GML files.

uDIG's UI is built on the Eclipse framework, so it will be familiar to anyone using the Eclipse software development environment. For non-programmers though, it can look a little complex at times.

The application has excellent support for styling, including supporting the OGC's SLD XML format for the look and feel of the map. This allows complex styles, such as linear features with borders, or dotted lines with customised sequences, to be created.

Unfortunately uDIG also has a number of bugs, which make the experience less than smooth. For example, after creating the map, the layers can become unavailable until the application is restarted. Layer styles sometimes disappear. When preparing a map for printing, the scale also tends to change without warning and can be completely inaccurate. Printing itself was not possible during tests, and the image export function is limited.

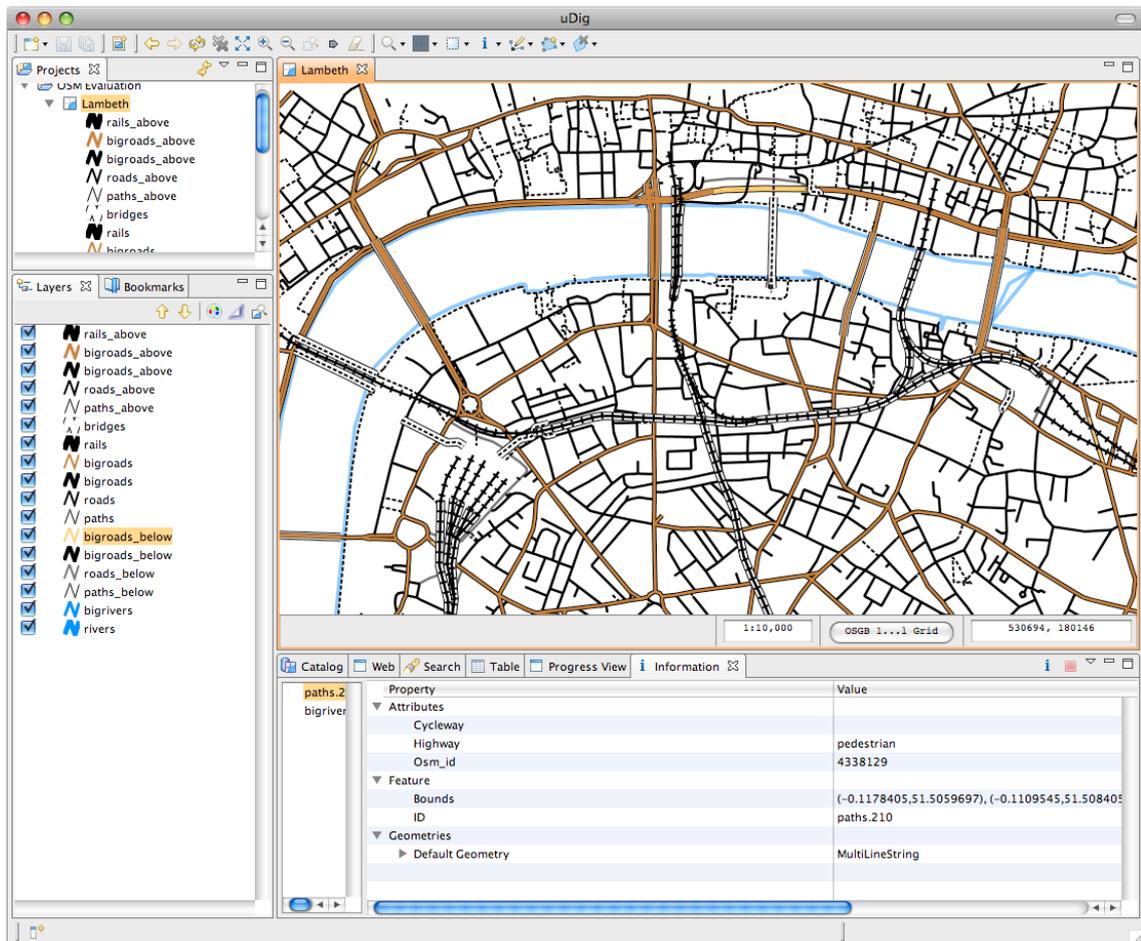


Figure 3-10: Preparing a Street-O map with the uDIG GIS.

3.4.4 OpenJUMP (1.2f)

OpenJUMP is cross-platform but, at least on the Mac, needs to be started from the command line. The interface is not very attractive or intuitive. It reads Shapefiles, and has limited support for GML, requiring a template file to be created to cross-map features. Its emphasis is on spatial analysis – for example, a built in function for calculating Thiessen polygons – rather than data presentation and visualisation, although it does support SLD files for styling.

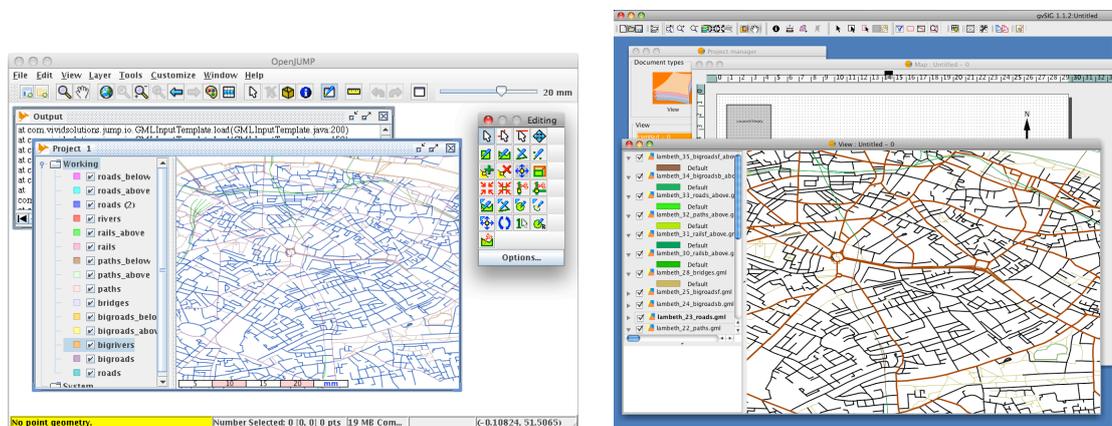


Figure 3-11: Left: OpenJUMP, Right: gvSIG.

3.4.5 gvSIG (1.1.2)

gvSIG imports DXF files (the only vector export format available in OCAD), Shapefiles and GML files. It has a clear split between the data view, tabular view and a map presentation view. The latter has a good number of features and offers plenty of customisation. However the GUI is very awkward and unintuitive to use. The English localisation is also currently very incomplete.

One interesting aspect is that a mobile version – gvSIG Mobile – is in current development.

3.4.6 Potlatch (0.10b)

This Flash-based application is designed for adding and editing information in the GIS database for OpenStreetMap (OSM). It was designed as a simple alternative for novice mappers to add map data – more established OSM mappers use JOSM, a Java-based offline application with a fuller feature-set but a steeper learning curve. Potlatch is very straightforward to use, and would be ideal for adding the points (nodes) and polylines (ways) needed for a Street-O map. Recently, a suggested tag taxonomy was added, complementing the free-form text approach that categorises OpenStreetMap. Some important features, such as levels (e.g. showing a railway-line crosses over a road rather than under it) require knowledge of the correct tags to use. The basic operation is not as intuitive as it could be, but once the few commands are learnt, maps can be edited and added to very quickly.

The main drawback is that Potlatch is tightly coupled with OSM and so could be used with it only. The OSM data is considered as one possible source of background data. Orienteering-specific data would have to be added and edited using a different editor.

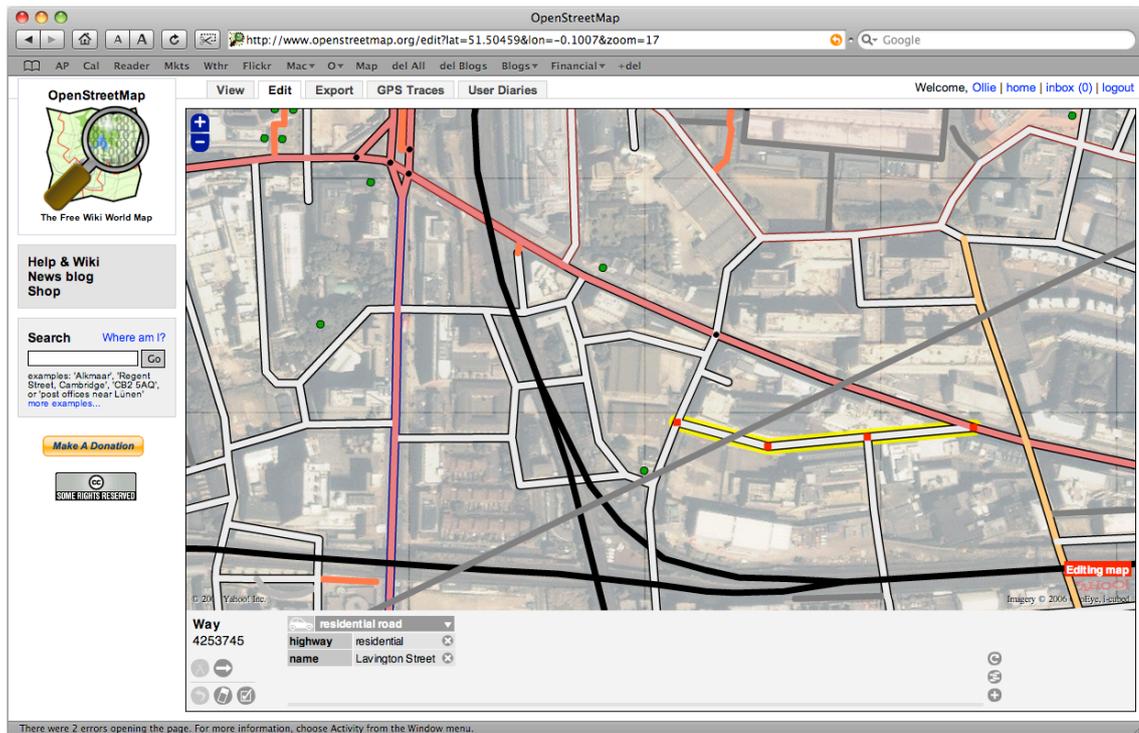


Figure 3-12: The Potlatch editor for OSM data.

3.4.7 Other OSM Editors

A number of other editors have been written for editing OSM data – JOSM and Merkaartor are two others currently in active development. These aren't considered here, as Potlatch is the default editor, accessed directly from the OSM website, and is straightforward and sufficient for the needs of this project.

3.5 Considerations in Choosing the Geo-Database

Ease of use and platform independence are not so important here as for the GIS application, because all communication with the server would be done through the GIS application itself, and because, with the typical simple client-server architecture model, there would only be a single server on a specific operating system, that could be connected to by different client applications.

3.5.1 Cost

In order to provide the complete end-to-end solution free-of-cost, the server would also need to be free. The support-based model that has become popular in recent years has allowed a couple of free-to-obtain database servers to be continued to be refined and improved, their development direction focused by their parent organisation, which benefits financially from providing support and bespoke installations to corporate users.

3.6 Review of Geo-Databases

In the free spatial database market, there are three main solutions – all three are implemented as extensions to the most popular free-to-use SQL database servers – Sun’s MySQL, PostgreSQL and Oracle’s XE. The extensions implement the OGC’s OpenGIS Simple Features Specification for SQL.

Each of the database servers has associated GUIs, developed to allow straightforward direct access to the data. For example, the PgAdmin application can be used to query data in a PostgreSQL database, including, by running the appropriate stored procedures, spatial data.

3.6.1 MySQL with Spatial Extensions

On its website, MySQL calls itself “The world’s most popular open source database” and is widely used for complex websites, due to its fast reading performance. MySQL is used for the OpenStreetMap database. MySQL Spatial Extensions add a number of geometry types and functions to represent and manipulate geographic features.

3.6.2 PostGIS (PostgreSQL)

PostgreSQL describes itself as “The world’s most advanced open source database”. The PostGIS extension, developed by Refrations Research, consists of various datatypes and stored procedures, allowing PostgreSQL to effectively store and deliver spatial data. Golicher (2008) states that “in the open source context ‘spatial database’ has become almost synonymous with the PostGIS component of PostgreSQL. It is the built in connectivity with PostGIS feature that gives QGIS and UDIG their rapidly expanding potential value.”

3.6.3 Oracle XE

Oracle describes its XE database as “an entry-level, small-footprint database... that’s free to develop, deploy and distribute”. Oracle Locator is a feature of the database, it includes geographic object types – a subset of the non-free Oracle Spatial extension, allowing the database to be used to store and query geographic data.

3.7 Considerations in Choosing the Base Map Source

3.7.1 Cost

Professional datasets typically have a cost – the “gold standard” for mapping in the UK is the Ordnance Survey’s MasterMap data, and other datasets it publishes, but these are expensive and with detailed licensing restrictions. Although the sport’s national governing body has an agreement (British Orienteering, 2008) with the Ordnance Survey allowing use

of its maps to be used as base maps for orienteering maps, there are various restrictions, such as requiring an additional license for each website a full-extent map is placed on. Moreover, as the agreement has to be renewed each year, the terms may change.

A source that is free-of-charge at source and unhindered by restrictions is therefore desirable.

3.7.2 Ease of Access

The base map data needs to be easily accessible to the mapper, ideally via the web, so that the mapper can quickly begin to create the orienteering map.

3.7.3 Quality of Data

The coverage, completeness within a covered area, spatial accuracy and taxonomic correctness (e.g. unambiguous indication of a real path as a path) of the base map data is important – the better the data quality, the less correcting will need to be done to it in order for it to be useable as a good base map.

3.8 Review of Base Map Sources

3.8.1 The Existing OCAD Maps

One possible source would be to convert and project the existing street orienteering maps, almost all of which are in the OCAD format – as the questionnaire above showed, this is the predominant application currently for creating Street-O maps. The maps cover a good area of central and southwest London, as can be seen in Figure 3-13, and have been drawn by mappers in the SLOW and LOK clubs. There are, however, a number of gaps between areas, which would need to be surveyed manually when creating a base map of a larger area.

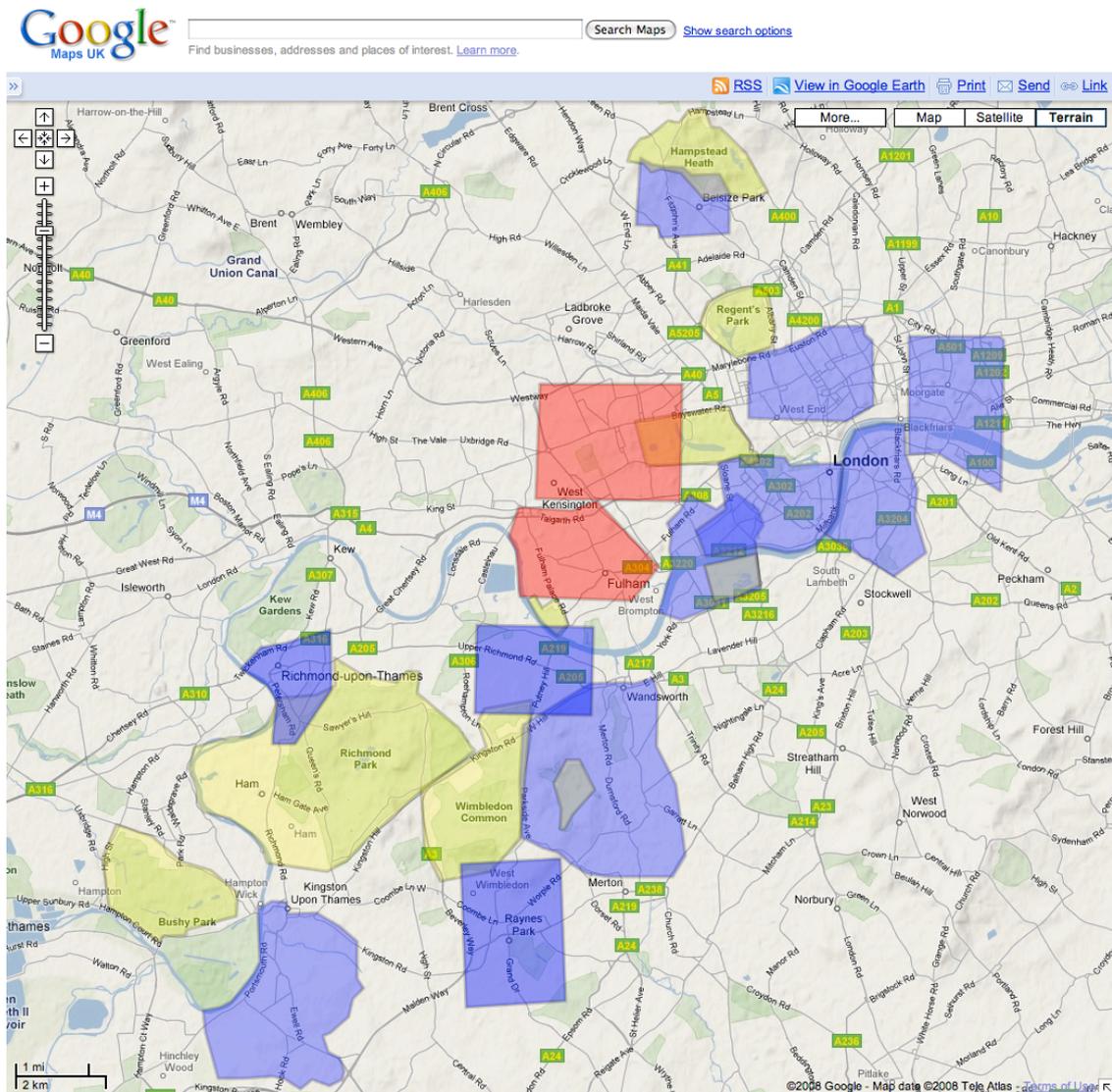


Figure 3-13: Areal extent of the known existing Street-O maps in central and southwest London. **Blue:** Areas for which Street-O vector data (generally in OCAD format) was obtained for this project. **Red:** Areas covered by Street-O maps but with no vector data available. **Yellow:** ISOM orienteering maps of parks.
 © 2008 Google – Map data © 2008 Tele Atlas.

A trial conversion procedure was carried out, with two of the map files – the Putney and Wimbledon maps. The maps were left untouched in OCAD. OCAD’s Export functionality was used to create DXF files. Initially, the option “Export OCAD curves as DXF splines” was ticked, however this resulted in none of the curved streets appearing in the corresponding DXF. The “Coordinates” option chosen was “GIS (m)”.

The Dxf2Shp plug-in in QGIS was used to convert the DXF files to Shapefile format. The DXF files had several layers present. Only the Polyline layer was exported, as, for the map files concerned, it was the layer that contained almost all the data.

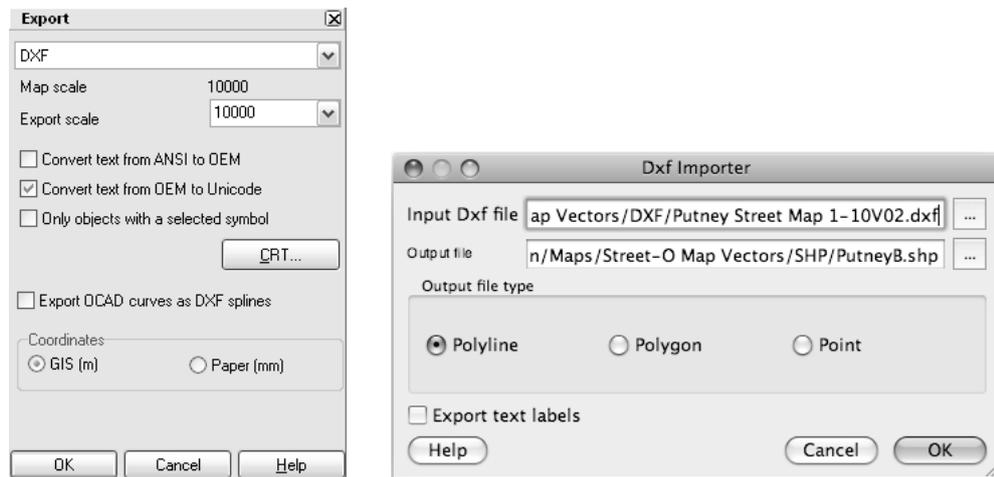


Figure 3-14: Converting the OCAD data to Shapefile format. **Left:** Using OCAD on a PC to export the OCD map files to DXF. **Right:** Using Dxf2Shp in QGIS to convert the polylines in the DXF file to a Shapefile.

The Shapefiles were then loaded into QGIS. It was assumed that the original files had been traced from a map using the British National Grid. Because the OCAD file coordinates have their origin at the centre of the map, it was necessary to create a custom map projection for each of the files. The approximate coordinate origin was located in QGIS and an Ordnance Survey grid reference was estimated, using Multimap. This was then converted to OS X and Y coordinates (e.g. for Putney, TQ236749, resulting in OS X 523650 and OS Y 174950) and these quantities were subtracted from the QGIS definition for the British National Grid projection.

QGIS uses the Proj4 system for specifying projections. The British National Grid projection, in Proj4 notation, is:

```
+proj=tmerc +lat_0=49 +lon_0=-2 +k=0.999601 +x_0=400000 +y_0=-100000 +ellps=airy +units=m +no_defs
```

In the case of the Putney shapefile, the custom projection was therefore set to be:

```
+proj=tmerc +lat_0=49 +lon_0=-2 +k=0.999601 +x_0=-123650 +y_0=-274950 +ellps=airy +units=m +no_defs
```

The Wimbledon and Putney map extents overlap slightly, which allows an easy comparison of the cartography. Unfortunately, the maps did not join up particularly well, with road alignments between the two maps not matching. Crucially the north direction appears to have been defined differently for the two – an indication that at least one had been rotated after tracing to point to magnetic north. It is convention for orienteering maps to be aligned to magnetic north, to allow runners to navigate without making the adjustment themselves. The declination varies non-linearly from year to year, and by location. Another possible cause is that the base maps used may also have different “grid north” directions – one may not have been using the British National Grid.



Figure 3-15: Shapefiles, converted from the OCAD data and georeferenced. **Red:** Putney. **Blue:** Wimbledon. The different northing declinations can be seen where they overlap. The legend and north arrow also appear on the Wimbledon shapefile.

Other problems which arose are the adornments appearing, such as the north arrow and partial legend in Figure 3-15, and that all the line features appear the same – the process of converting them loses the distinguishing information which would allow paths, roads and railways to be distinguished differently in a map produced from the data.

While the OCAD map data is therefore not directly usable, it is still useful as a check for any other base map data source used, to ensure that the correct features are being used and that the completeness of the source is at least equivalent to the OCAD maps.

3.8.2 Raster Tracing

The questionnaire showed that the existing maps are generally traced from raster images, typically from mapping websites, or a scan of a street-map – OCAD allows a raster to be layered in the background, with features added based on it. However several respondents said that this was the most tedious part of the process.

The rasters themselves can be subject to distortion (particularly if the image is scanned) or generalisation by the organisation producing them – detail could be lost or distorted, simplified or displaced for cartographic reasons, benefiting most general-purpose users of the map but perhaps not orienteers. It is normally difficult to spot such distortions, even when field-checking the map.



Figure 3-16: Example of cartographic displacement on a raster Street-map. **Left:** Extract from the Collins Bartholomew imagery, as used on the Multimap website, for an area in Hertford. © 2007 Collins Bartholomew Ltd. **Right:** The equivalent area, as shown by OpenStreetMap's Mapnik tile renderer.

Figure 3-16 shows an example of cartographic displacement for raster maps. The smaller “Port Hill” road’s south end starts just to the west of the car park access road in the Collins raster, and to the east in OSM – the latter is spatially accurate, but at smaller scales, the road is too close to the larger road, also called “Port Hill”, to be shown in the correct position. An orienteer leaving the car park access road, using a map traced from the Collins raster, might turn the wrong way when trying to find the smaller Port Hill.

3.8.3 OpenStreetMap

The OpenStreetMap (OSM) project is aiming to map the entire world, with volunteers adding new features by tracing GPS routes, aerial photographs, out-of-copyright maps and personal knowledge.

Most of London is now nearly complete – certainly coverage of Central London is good, and weekly “mapping parties” have been held throughout Summer 2008 to check and enhance more suburban areas (MacDonald, 2008). It is a matter of some debate within the community whether an area such as London can ever be declared “complete”, as people have different measures of completeness, the city is changing and evolving, and there is no fixed list of features that must be present to complete an area.

Several studies have recently compared the quality and completeness of OSM data with that from other mapping providers (Grant, 2008; Haklay, 2008b). Haklay’s study compared the accuracy and completeness of OSM data with Ordnance Survey’s Meridian dataset, which can reasonably be considered to be the definitive standard. The study shows that, for much of the London area, OSM’s coverage, of the road network at least, is excellent, helped in part by the large number of contributors in the London area – the project indeed started in London.

OpenStreetMap offers a simple interface to extract the data behind the rendered maps on its website, via a webpage where an area of data desired can be downloaded. Additionally the entire data-set can be downloaded, as a “planet.osm” file, currently over 4GB of compressed XML and updated on a weekly basis, with regional subsets also available.

A number of technical problems would need to be overcome, for example the display of bridges. The OCAD maps use distinctive polylines to make the bridges very visible, such clarity would need to be derived from the OSM data that is unlikely to exaggerate such features.

3.9 Visual Comparison of OpenStreetMap and Existing OCAD Data

To assess the completeness of OSM data, against with the data from the previously produced maps in OCAD, an exercise was carried out, for the Putney area.

Because the OCAD map, which was assumed to have been traced from a British National Grid-projected map, had then been rotated, it was not possible to use it easily as a vector. Instead, it was rasterised and QGIS’s Georeferencer plug-in was used to line it up with the OSM data layer and create a new GeoTIFF raster, effectively rotating back to the British National Grid and allowing the two maps to be superimposed.



Figure 3-17: Graphical difference between the original Street-O map and the OSM data. **Blue:** The original Street-O map. **Red:** A subset of features from OSM that would appear on a Street-O map produced with it. Also shown

are some possible significant missing information on the original map (**Orange Markers**) and the OSM data (**Green Markers**)

A number of cul-de-sacs are present on one map and not the other – this does not significantly affect the quality of either map, as the runner would not use a dead-end unless a control was placed at the end of it. Detail on the edges is also ignored due to differences between the exact area chosen for the two maps, and the “hairy” nature of the OSM data – this is where polyline (and areal) features are partially present in the rectangle of data selected for use, but also extend outside it. They are included in their entirety in the imported data, resulting in “hairy” strands appearing on the edges.

More significant discrepancies between the two maps are annotated. These are:

Possible missing features from original map:

- A. This bridge over the railway is missing.
- B. The road is shown as a cul-de-sac rather than a loop here.
- C. This section of railway is missing altogether.
- D. This possible optimal route between two roads is missing.

Possible missing features from OSM data:

- 1. The path connections from the estate to the road to the south are missing.
- 2. A lot of minor roads and paths between the roads are missing.
- 3. Only three cul-de-sacs are shown in this cluster, rather than four.
- 4. This connecting road is missing.

Overall, the two data-sets generally match very well, neither seems significantly more complete or detailed, and a race run using either set is unlikely to significantly cause problems for competitors due to numerous vital shortcuts being missed.

MasterMap from the Ordnance Survey, including the Integrated Transport Network (ITN) layer and a restricted feature-set from the topography layer, was also considered for comparison, as a professionally surveyed, effectively definitive source of data. However the ITN does not include paths, and the “Roads, Tracks and Paths” feature-set in the topography layers is too detailed for orienteering, for example showing private driveways behind security gates, and road/pavement edges.

3.10 Considerations in Producing the Map

While a GIS will manage the spatial data, potential control points and metadata, the ultimate purpose is to produce an orienteering map, so it is necessary to consider how the

final map will be produced. Ideally, it should look as clean, simple and easy-to-read as the existing Street-O maps. Street-O is an “informal” style – unlike the ISOM and ISSOM international standards for “normal” orienteering maps, the Street-O style is not defined anywhere. This project concentrates on Street-O in the London area, so aims to produce maps that mimic the designs used here. Typically, these show solid black roads, dashed black paths, simple bridges and railway lines, if desired, and major rivers in blue. The control points themselves appear in magenta overprint, which mimics normal orienteering maps.

3.11 Review of Techniques to Produce the Map

3.11.1 Customising the PDF Atlas Utility in OpenStreetMap

White (2006) has produced a method of directly producing maps from OpenStreetMap data, in PDF format for easy printing, using the Perl scripting language. Configuration scripts allow a specified set of features only (e.g. all roads) to appear on the map, define the desired extent of the map and colours for each feature.

It should be possible to adopt the script for printing Street-O maps. However, the lack of a GUI would mean familiarity with the command line would be needed to use an adapted script, for example control points would need to be specified in an additional file. Further development would be needed to use data from non-OSM sources.

3.11.2 Exporting OpenStreetMap data as SVG using OSMARender

The data can be exported as SVG, a highly flexible format for describing vector graphics. It should be possible to programmatically apply styling, possibly specified in the standard SLD format, to an SVG or import it into a dedicated CAD or vector graphics program, particularly an SVG-based one such as Inkscape, to complete the map. This approach doesn't make use of a GIS so the benefits of layering spatial data from other sources, a key ability with GIS applications, would not be realised.

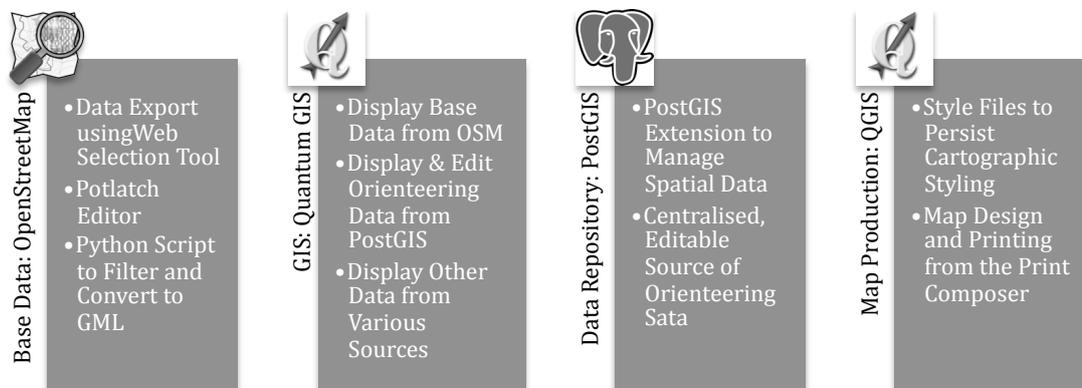
3.11.3 Direct printing from the GIS

The GIS applications reviewed generally have a mode allowing composition of a simple map for printing – not with the advanced cartographic capabilities of CAD and graphics applications, but sufficient for the level of detail required for a Street-O map. QGIS has a Print Composer component, which is reasonably flexible and allows placement of the map, text, images, legends and scale bars.

4 Design

The chosen solution, for creating and maintaining Street-O maps using a GIS, is:

- OpenStreetMap data as a base map, with a customised Python script to filter and split out desired layers and convert them to GML, with the web-based Potlatch editor to edit base map features.
- Quantum GIS (QGIS) for viewing various datasets and editing orienteering data.
- PostgreSQL with PostGIS for storing the control points and other orienteering data.
- QGIS's Map Composer and QML style files for creating a printable map.



4.1 The Base Map

The OpenStreetMap data, at least for central London and other urban areas with an active community of OSM mappers, was determined, during the analysis above, to be suitable for use as a base layer for Street-O maps.

4.1.1 Obtaining the OSM Data

The three main ways of retrieving the data are from a database-dumped “planet.osm” file, from a web service “Osmxapi”, or from the OSM website’s XML download page. The Osmxapi service allows for tag filtering at the point of query. However, for simplicity, the latter approach is used.

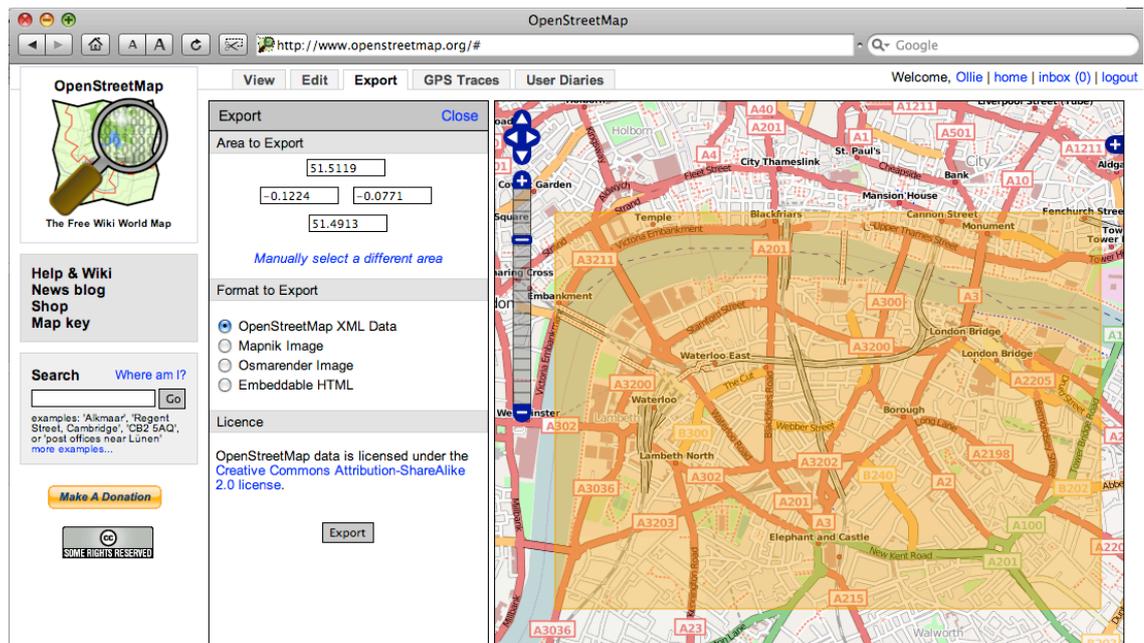


Figure 4-1: Exporting the data from the OSM webpage.

A latitude/longitude extent can be specified or an area specified. The XML file retrieved contains the latest data from the OSM database, which may be more recent than what is seen in the weekly-rendered rasters on the website.

4.1.2 Filtering and Converting the Data

Once the data is retrieved, the next step is to remove the data that is not of interest, and convert from OSM XML to GML format, suitable for importing into the GIS application, QGIS.

OpenStreetMap does not have a fixed taxonomy, a deliberate decision to encourage growth of the mapping across the world without restriction. Recently however, an effort has been made to standardise the most common features. The Potlatch editor now has a suggested, but non-binding, taxonomy, accessed via drop-down lists that appear when starting to assign tags to features.

From studying the SLOW and LOK Street-O maps, tags attached to appropriate features in the OSM Potlatch editor, and the OSM wiki's Map Features page, it was decided that the following features are needed from OSM in a Street-O map. It should be noted that the Map Features page is not exhaustive and other tags can be present.

- Roads – major roads and other roads as separate layers.
- Paths and cycleways.
- Rivers, streams and lakes (optional).
- Railway lines (optional).

The “Tagwatch Great_britain Undocumented Keys/Tags” pages on the OSM wiki were used to confirm that no further relevant tags than those in the Map Features page were used in significant quantities in the UK for Street-O-useful features.

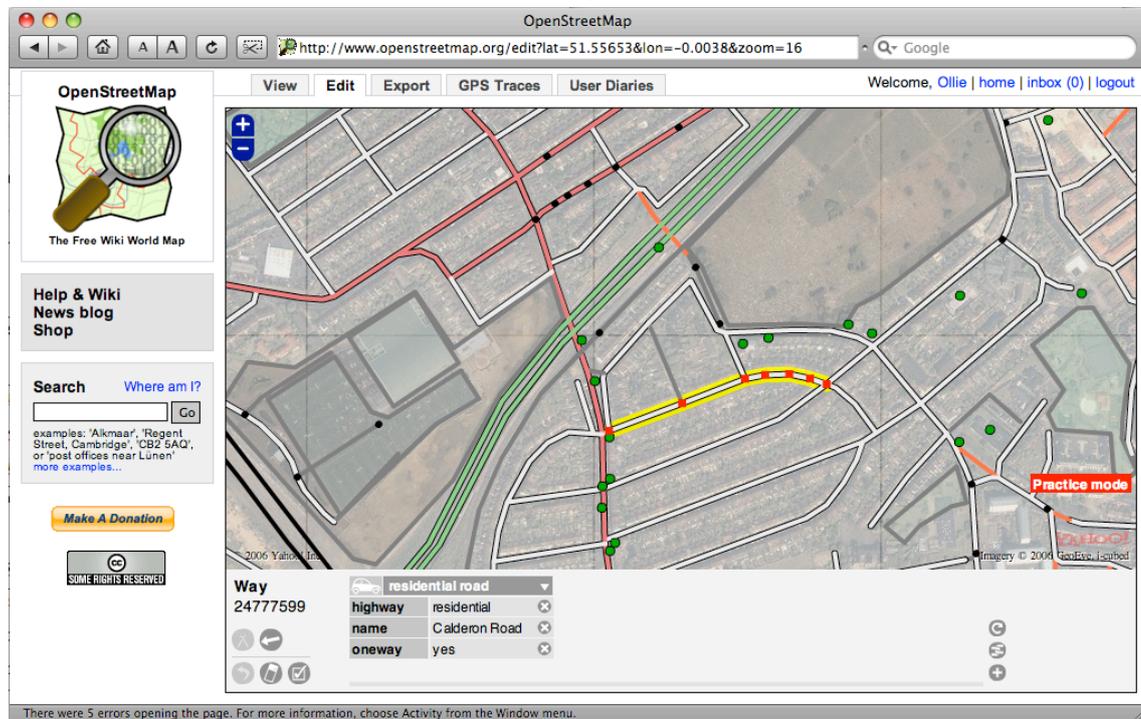


Figure 4-2: Viewing tags associated with a way in Potlatch.
The aerial imagery is supplied by Yahoo and is ©2006 NAVTEQ. All rights reserved.

In OSM, features have attributes, known as tags. There can be several of these, made up of non-hierarchical key/value pairs.

Looking at the tags attached to appropriate features (the ones that generally matched up with those on the existing Street-O maps) it is believed that features that contain at least the tags in Table 4-1, would want to be retrieved.

Keys and values in the Notes column appear as key=value. Values in italics were not seen in a cursory examination of a sample area of central London, but are listed in the OSM map feature guide. Key/value pairs are in theory additive, e.g. there could be multiple values for the highway key, or large rivers could be tagged with one or both of waterway=riverbank and natural=coastline.

Street-O Feature	OSM Key	OSM Values	Notes
Major roads	highway	<i>motorway</i> , <i>motorway_link</i> , <i>trunk</i> , <i>trunk_link</i> , <i>primary</i> , <i>primary_link</i> , <i>secondary</i>	
Roads	highway	<i>tertiary</i> , <i>unclassified</i> , <i>road</i> , <i>residential</i> , <i>living_street</i> , <i>service</i> , <i>bus_guideway</i>	Future development could exclude ways where highway=construction.

Paths and Cycleways	highway	track, pedestrian, path, cycleway, footway, bridleway, byway, steps	
	cycleway	Track	
Rivers	waterway	riverbank, dock	Boundary of an areal feature
	waterway	canal, river, stream, drain	A linear feature
	natural	water, coastline, river, lake	Boundary of an areal feature
Railway lines	railway	rail, light_rail, subway, tram	Not desired if tunnel=yes
Bridges	bridge	yes, true	The bridge key is also used as a modifying attribute for other features.

Table 4-1: OSM tag keys and values corresponding to desired street-O map features.

Street-O Attribute	OSM Key	OSM Values	Notes
Levels	layer	-5, -4, -3, -2, -1, 1, 2, 3, 4, 5	Layer is an integer from -5 to 5, with 0 corresponding to ground level.
Tunnels	tunnel	yes, true	Value ignored except if false or no.
Bridges	bridge	yes, true	We set the level to 1 if it is not specified but the feature is tagged as being on a bridge.

Table 4-2: OSM tag keys and values for additional useful street-O feature attributes that may be present.

In order to show bridges correctly in the map, and hide underground railways, a further three keys, which may be present on the features in the first table, will need to be retrieved – bridge, layer and tunnel.

OSM doesn't store polygons – instead, closed polylines, called ways, are rendered as areas for the raster tiles. Coastlines are a special case due to their length. They are not closed – the direction of the polyline is used to infer and correctly render the “sea” side.

This means it is not possible to directly pull in areal features into QGIS, from OSM. The only areal features this project is concerned with, for the area studied, are large rivers and lakes – in London, the River Thames is the main areal water feature. It is relatively straightforward to convert a GML polyline into a polygon, so is done during the conversion process for the appropriate layers.

4.1.3 Editing the Base Map

It is intended that the base map contains all the features required for the final map, except for those specific to Street-O – the control sites, the start/finish point, map adornments, special instructions and out-of-bounds areas. Instead of adding or modifying non-orienting features in an orienting map database, it is intended that such features be directly edited at source.

Editing at source has the advantage of the work benefiting the OSM community at large, allowing the management of the data to be globally accessible to all, including future

Street-O mappers, and benefiting from updates and corrections made by other people. The disadvantages are that map vandalism – not a major problem on OSM yet – may be inadvertently picked up, or well-meaning non-orienteeing editors might remove certain real-world features they consider not to be notable but which would be useful to a Street-O competitor, such as small paths shortcutting a road corner.

Care must be taken not to add data derived from copyrighted sources, e.g. Ordnance Survey maps. However data can still be entered without visiting the area, as for the London area, the Potlatch editor displays recent aerial imagery for tracing, from Yahoo. Viewing the images can often resolve queries such as whether a railway passes over or under a road.

While an alternative approach of copying the OSM data in an orienteeing-specific database and pulling updates across regularly allows for more control over the base map, it does increase the system's complexity, and would lose either the real-time nature of the updates (if the edit was done in the OSM database) or the benefit to OSM (if the edit was done in the orienteeing-specific database).

Potlatch is intended to be the editor used. Although it is not as powerful for GIS-familiar users, it is easier to use. Its functionality is limited but it has the major benefit of a low entry barrier – all that is needed is a Flash-enabled web browser and Internet connectivity. It is in active development and gaining new functionality with each release, and is accessed directly from the map on the OSM website.

One important aspect of editing in OSM is that, when switching away from the editor to view the main map, the raster “tiles” are not re-rendered immediately. However, the edits are committed immediately to the OSM database and so can be instantly retrieved.

4.2 The GIS

A large number of free-to-use GIS applications were considered above. In terms of the needs for this project – importing the data, styling it and preparing a map in a straightforward and intuitive way – UDIG and QGIS stand out as being the most complete and “polished” applications. QGIS will be used for this project, as its straightforward, simple design should make it readily useable by Street-O mappers.

4.3 Managing Orienteering-Specific Data

Orienteering-specific data will be managed in a PostgreSQL database with the PostGIS extension installed, which has excellent integration with QGIS. If the geo-database is hosted in a central location, this will allow the data to be shared across different mappers

without needing to keep track of a peer-distributed set of files, while also keeping the orienteering data away from general-purpose datasets like those in OpenStreetMap.

4.3.1 Schema Design

If it is the first time that the database is being used, then it is necessary to set up the tables needed. This can be done by creating a Shapefile with the desired attributes in QGIS, adding at least one feature (which can be deleted later once “real” features are present) and converting it to a PostGIS table by using the Shapefile to PostGIS Import Tool (SPIT), a plug-in that comes with QGIS. The set of attribute datatypes available when creating the shapefile is limited, but sufficient for these purposes.

Four tables are needed in the database for the orienteering information. As well as the attributes listed below, two more get created automatically for each table during the conversion process – “gid” which is the primary key and appears as “id” when viewing the attributes in QGIS, and “the_geom” which contains the spatial geometry for each feature.

Controls – these are the places on the race where competitors are going between. They are point features and there may be as many as 40 for each race.

Column Name	QGIS Datatype	Required?	Purpose
event_name	String	Yes	Planned date of the event.
control_no	Integer	Yes	The control number on the map, normally between 10 and 50. It is displayed as the feature’s label.
score	Integer	Yes	The score associated with each control, defaults to 10.
clue_q	String	Yes	The question that the competitor must answer to get the points for this control. For example, “Which artist lived at No. 26?”
clue_a	String	Yes	The answer. For example, “Shepherd.”
label_x_of	Integer	No	Can be used to displace the label if it is obscuring other features.
label_y_of	Integer	No	Can be used to displace the label if it is obscuring other features.

Start & Finish – the place where the race starts and finishes. It is a point feature and there is only one for each race.

Column Name	QGIS Datatype	Required?	Purpose
event_name	String	Yes	Name of the event, e.g. “Putney April 2008”

Out-of-Bounds – Areas where competitors are not allowed to run, because it is too dangerous for them to do so, such as near steep riverbanks, or because the area concerned is not mapped. Very busy roads may also be marked with an out-of-bounds area, even though the feature concerned is linear, to add emphasis – or a similar table containing linear features could be created.

Column Name	QGIS Datatype	Required?	Purpose
event_name	String	Yes	Name of the event, e.g. “Putney April 2008”
label	String	No	Information to be displayed over or alongside the shaded area.
additional	String	No	Any further information that may be of use to a planner.

label_x_of	Integer	No	Can be used to displace the label if it is obscuring other features.
label_y_of	Integer	No	Can be used to displace the label if it is obscuring other features.

Annotations – Notes for the current and future mappers and planners of the area, for example advising on suitable start points, contact details if permissions are required for parts of the map, expected street-closures for road-works, and information on park gate closing times. For simplicity, point features are used here, but a similar table containing linear or areal features could be used instead.

Column Name	QGIS Datatype	Required?	Purpose
label	String	No	An optional short message that would be displayed directly in the view for the mapper.
annotation	String	Yes	The main information for this annotation, recording information about the point (or nearby street or area) being flagged. It may be useful to record the date on which the annotation was noted, e.g. the date that a path was closed for rebuilding, or the date of the event if the issue only applies to that particular event.

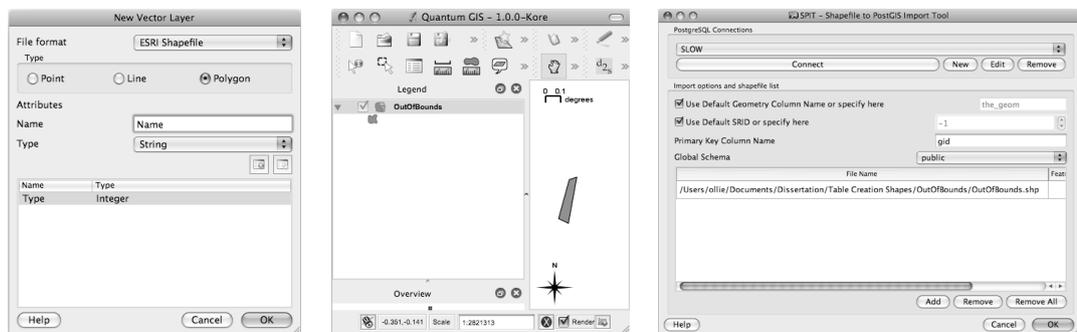


Figure 4-3: The three-step process in adding a new PostGIS table for orienteering data, using QGIS. **Left:** Creating a new shapefile. **Middle:** Adding a test feature to the shapefile. **Right:** Using SPIT to convert the shapefile to a table on the PostGIS database.

The SQL table schemas can be seen in Appendix D.

4.4 Producing a Map for Orienteering

4.4.1 Symbol Design

QGIS allows custom designs for point symbols to be used, if they are in SVG format and placed in the appropriate directory included with the application. However, the built-in designs are sufficient for symbols appearing on Street-O maps.

4.4.2 Styling

Each layer can be given a cartographical style, by setting various options in the properties for the layer in QGIS.

As an alternative to continuously recreating each style, QML style files can be loaded for each layer, to give them an appearance similar to that of the OCAD maps. The style files can be created in QGIS by saving the current styling for a layer. The layer styling is based on the

ISOM specification, but with slightly wider strokes for clarity on screen and print, as the low-key nature of Street-O events means maps are likely to be printed on lower-quality printers than regular ISOM maps. The colour picker tool in QGIS can be used to build up the correct colours. Loading in the style file for each layer also causes the layer name to change, to prepare it for use in the legend for when the map is being printed. (QGIS currently has a bug where style files do not work across different projects, so the styles were recreated for each case study and QML files were not used.)

Feature	Stroke (mm)	Colour	C (%)	M (%)	Y (%)	K (%)
Major Road	0.6	Light Brown	0	28	50	9
Major Road (Behind)	1	Black	0	0	0	100
Road	0.5	Black	0	0	0	100
Path	0.25	Black	0	0	0	100
Water (Areal)	0	Light Blue	44	9	0	0
Water (Linear)	1.5	Light Blue	44	9	0	0
Railway	1	Grey	0	0	0	50
Bridge	1.5	White	0	0	0	0
Bridge (Behind)	2	Black	0	0	0	100
Contour	0.14	Brown	0	56	100	18
Index Contour	0.25	Brown	0	56	100	18
Control	0.5	Magenta	0	100	0	0
Start and Finish	1	Magenta	0	100	0	0
Out of Bounds (Areal)	0	Magenta	0	100	0	0

Table 4-3: Suggested QGIS settings for styling used for map features.

QGIS doesn't support the OGC's SLD standard for setting how a feature should look. Should support for SLDs be added in the future, more advanced styling will be possible. This would, for example, negate the need to have additional layer files used to "background" a stroke, as SLDs allow for multiple strokes per polyline layer.

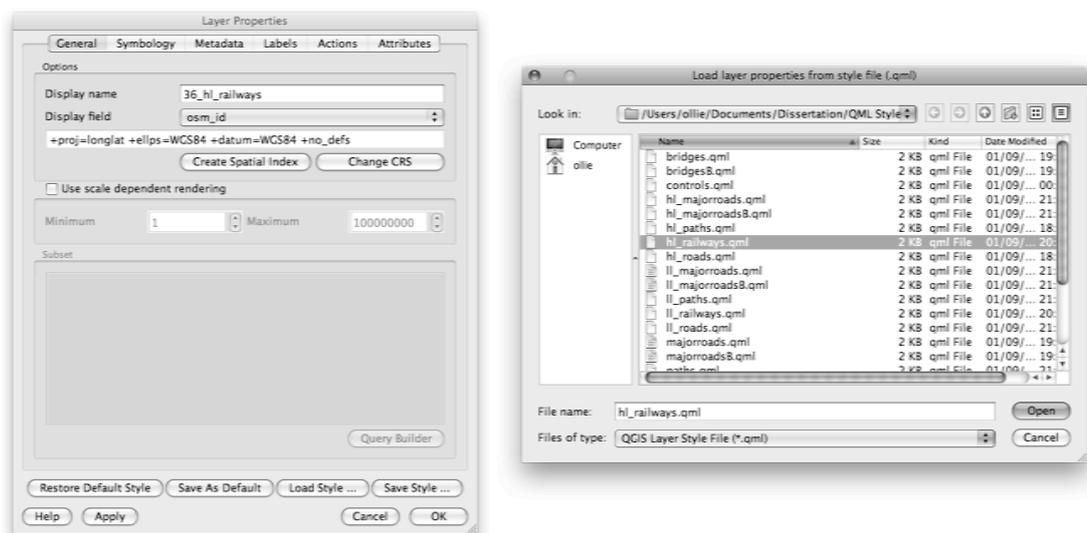


Figure 4-4: Applying a style file.

Each style file needs to be applied manually, although QGIS's support for control by python programming means that this step may be able to be automated.

4.4.3 Print Composer

QGIS's print composer contains sufficient functionality for producing a Street-O map, with a comparable style and collection of adornments to existing London Street-O maps. It is highly configurable and can output directly to a printer or to SVG format.

5 Implementation

5.1 Setting up the Software

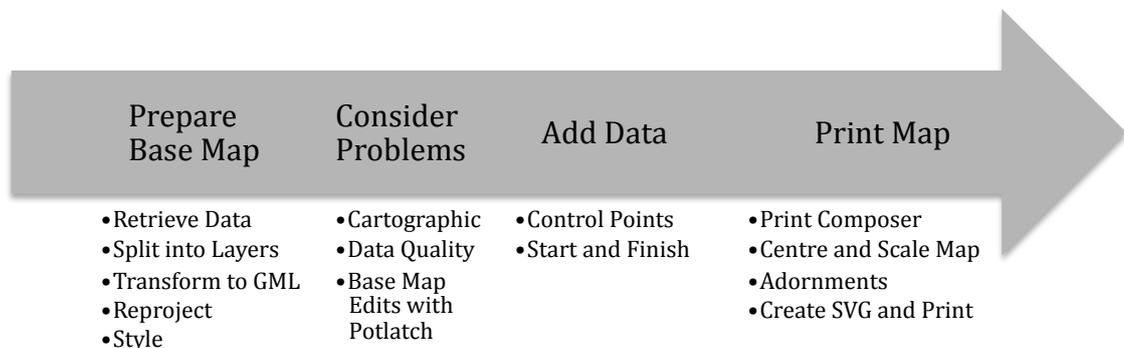
The software was installed on a computer running the Mac OS X operating system, which acted as both the server to deliver data from the geo-database, and the client, on which the QGIS application was run. In a real system, the two functions would normally be on two separate computers, with the geo-database accessible online to the mappers.

The PostgreSQL database, the PostGIS and PgRouting extensions, and the PROJ and GEOS frameworks were installed on a test server running Mac OS X, from pre-built packages (Kyngesburye, 2008). The PgAdmin application was also installed to help with configuring and observing the database. QGIS was downloaded from www.qgis.org and installed.

5.2 Introduction to the Map Case Studies

The design was implemented by two case studies, as examples of use of the GIS for Street-O orienteering. The first case study looks at the end-to-end process of creating a Street-O map using a GIS. The second one focuses on adding various datasets from different sources, to enhance the map.

5.3 Case Study 1: Putney



Dorte Torpe Hansen of SLOW originally drew a Putney Street-O map in OCAD. It was last used for a Street-O event in March 2008. This study uses the OpenStreetMap base map data, and QGIS, to recreate the map, allowing a comparison of the cartography of the old and new techniques.

The styles on the OCAD map do not match completely with those in the other SLOW Street-O maps – for instance, all roads, including the major roads, are shown with equal-width black lines, and the River Thames is shown only thin black lines representing the banks – as

can be seen in Appendix F. Styles that match the majority of the existing maps were used so that all maps produced with the new technique have consistent styling.

5.3.1 Obtaining the Base Map Data

The extent of the old map was estimated visually in OpenStreetMap, the area was located and the data extracted in OSM XML format using the web interface, to a file, renamed Putney.xml, which was around 0.8 MB.

The web interface is used to select an x-y rectangular bounding box, and it effectively requests the following URL that supplies the data file:

```
http://api.openstreetmap.org/api/0.5/map?bbox=-0.24097,51.45087,-0.20196,51.46935
```

The area selected needs to be slightly larger than the extent of the old map, as changing coordinate reference system from that associated with WGS 84 (used by OSM) to the British National Grid effectively results in a slight rotation, but the area selectable when creating the map still needs to be x-y rectangular.

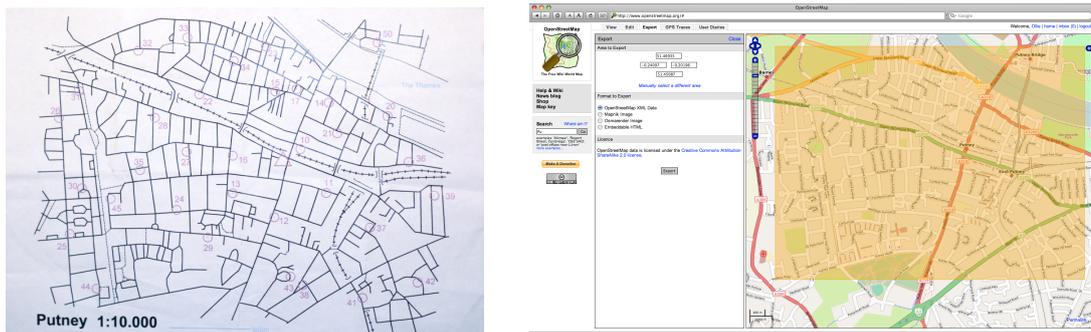


Figure 5-1: Left: The old Street-O map produced in OCAD. A full-size copy can be seen in Appendix F. Right: Selecting the equivalent area on the OpenStreetMap website.

The file was then processed by a python script, an extensively modified version of osm2gml.py (Zeitz, 2008; Erle, 2006), which extracted the features listed in Table 4-1 and Table 4-2 into separate files, each representing a layer, and converted them to the GML format. The script converts polylines for water edges into polygons.

The modified version of osm2gml.py can be seen in Appendix B.

The set of arguments used to process the Putney.xml file with osm2gml.py were:

```
python osm2gml.py Putney.xml layers
```

5.3.2 Levels and Layers

OpenStreetMap features can be tagged with a level number (the key name in OpenStreetMap is “layer” but “level” is used here to distinguish from the layer files) to indicate whether they are above or below other crossing features.

In order to display bridges and tunnels correctly, while minimising the number of layer files needed, a simplification is made – all features with a level of 1 or greater is assumed to be above the main road level, or high-level (hl) while all features with a level of -1 or less are assumed to be below it, or low-level (ll). This means that very complex junctions with more than three levels overlapping will not show properly – it is assumed that such junctions are rare in the terrain that Street-O races would expect to be held on.

Data marked as being part of a bridge but without a level specified is assumed to be above the main road level (this is separate to the specific bridge layer.) Railways are assumed to be above roads where there is no level distinction – this appears to be how the OpenStreetMap renderers themselves display such overlaps. Water features are assumed to always be below everything else. Tunnels are shown for runnable features (roads, paths) but not railways.

Note that being below the main road level does not necessarily mean a tunnel – for example, a railway in a cutting would probably be recorded in OpenStreetMap with a level of -1 (simplified to low-level here) because the surrounding land is above it, even though it is open to the air and so should be shown on the map.

Major road and bridge layers are generated twice – one layer is used as an extra stroke in the background to mimic the cartography in most of the OCAD Street-O maps – as QGIS is currently limited to only showing one stroke per line.

Bridges are produced in a separate layer – these are placed under the above ground levels to emphasise them.

The script contains the following layer files – listed here in the order in which they are intended to be displayed in QGIS, the first being the layer displayed on top. The numeric prefixes ensure the layers are displayed in the intended order when bulk-added to QGIS, as it adds files sequentially in alphabetical order, from the bottom up. This table should be compared with the desired layers from OpenStreetMap in Table 4-1 and Table 4-2.

Layer file (.gml)	Legend Caption	Level	Shown if in Tunnel?	QML Style File Name (.qml)	Number of Features on Putney Map
36_hl_railways		Above	No	Railways	4
35_hl_majorroads		Above	Yes	Majorroads	1
34_hl_majorroadsB		Above	Yes	MajorroadsB	1
33_hl_roads		Above	Yes	Roads	1
32_hl_paths		Above	Yes	Paths	2
28_bridges		Above	No	Bridges	6
27_bridgesB		Above	No	BridgesB	6
26_railways	Railway	Ground	No	Railways	4
25_majorroads	Major Road	Ground	Yes	Majorroads	31
24_majorroadsB		Ground	Yes	MajorroadsB	31

23_roads	Road	Ground	Yes	Roads	354
22_paths	Path	Ground	Yes	Paths	129
16_ll_railways		Below	No	Railways	1
15_ll_majorroads		Below	Yes	Majorroads	0
14_ll_majorroadsB		Below	Yes	majorroadsB	0
13_ll_roads		Below	Yes	Roads	0
12_ll_paths		Below	Yes	Paths	1
02_waterareas	Water	Below	No	Waterareas	1
01_waterlines		Below	No	Waterlines	1

Table 5-1: Table of GML layer files created by osm2gml.py for use in QGIS to create a Street-O map. The layer file prefixes and suffixes are: hl – high level, ll – low level, B – background stroke layer.

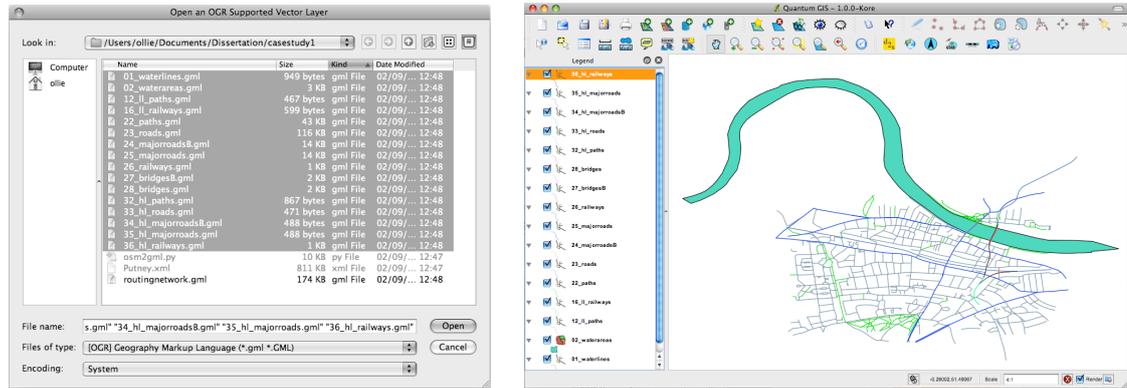


Figure 5-2: Left: Bulk-adding the GML layer files. Right: The layer files displayed in QGIS, before projection information and styling is applied – the default colours are randomly chosen by the application.

5.3.3 Preparing the Base Map

Once the files are bulk-added to a QGIS project, it is necessary to configure the QGIS project. Firstly, a suitable projection needs to be applied. Although the OpenStreetMap data and the resulting GML files are in the commonly used WGS 84 datum's standard projection; OCAD Street-O maps, and indeed most maps centred on UK areas, are in the British National Grid projection. This projection minimises distance distortions and is approximately planar at large scales in the UK. WGS 84's geographic coordinate system, in the London area, causes the data to be distorted horizontally by a factor of around 1.73 – this is the difference between latitudinal and longitudinal degree distances here.

The project itself can be defined to be in the British National Grid projection. Additionally, the map units can be set to be in metres. A project name can be entered here and the “on-the-fly” checkbox is ticked so that the re-projection takes place. These settings are all configured in the Project Properties dialog, shown in Figure 5-3.

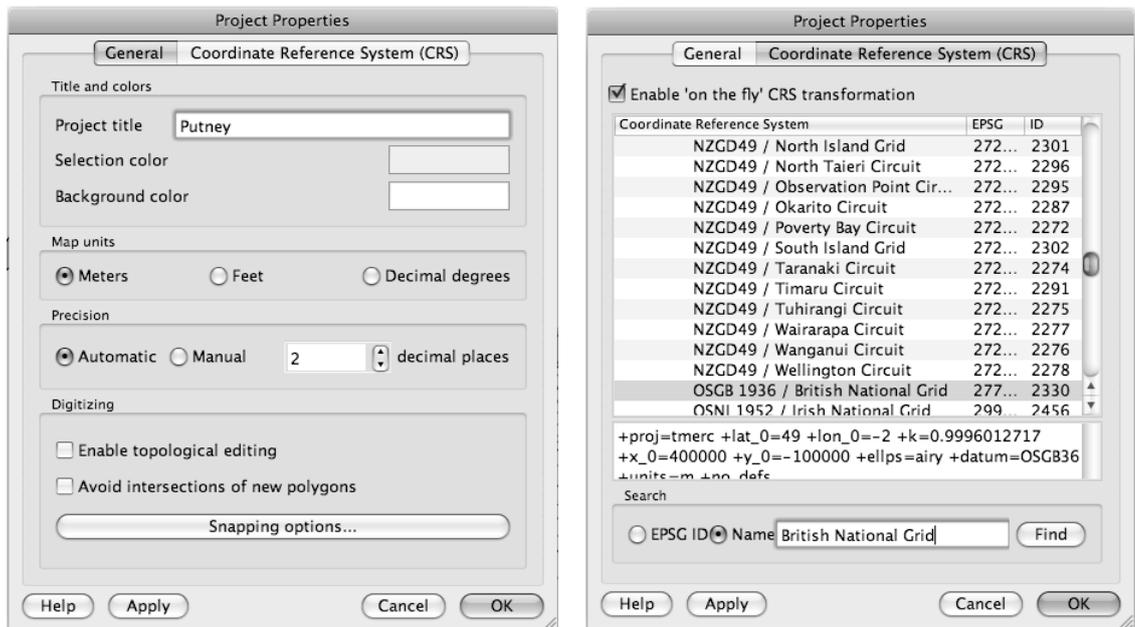


Figure 5-3: The Project Properties dialog, with options set for correct display of the data in QGIS.

The layers are then styled, as specified in the design. Style files are saved for possible future re-use.

Finally, typing in the desired scale for the view, in the box at the bottom of the project window, will set the scale appropriately. The view now starts to resemble a Street-O map.

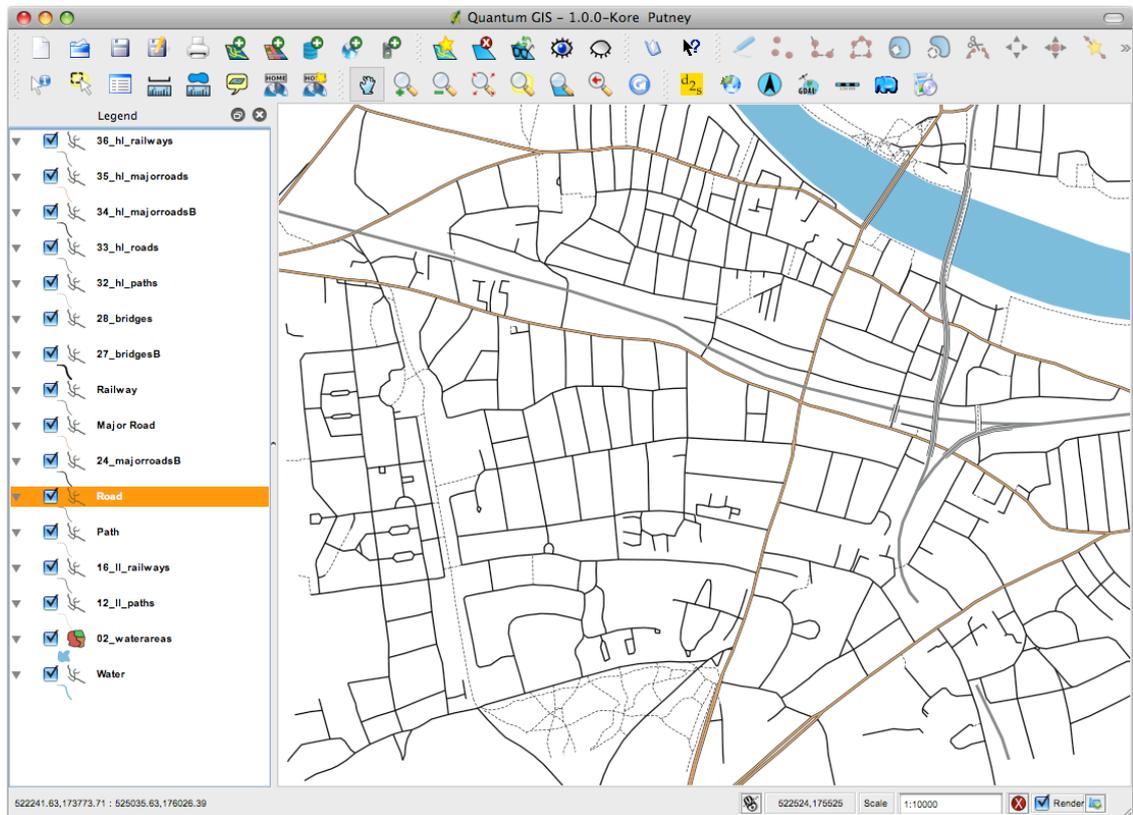


Figure 5-4: The new Putney map in QGIS, after the styling is applied and the view re-projected and scaled.

5.3.4 Cartographic Problems

1. The cap for strokes cannot be easily controlled in QGIS, unlike in uDIG, so the ends of bridged polyline sections (e.g. roads on bridges) have an white and black section, the protruding cap from the wider bridge layers apparently severing them from the adjoining surface polylines, as shown in Figure 5-5. A small patch for QGIS was written and compiled into the application (Appendix C), setting the default to be “flat cap” rather than “square cap”, to eliminate this problem. It is hoped that configurable capping will eventually be included in the QGIS application.
2. The footbridge beside the railway bridge across the river is nearly obscured by the railway stroke, at the 1:10000 scale. Reducing the stroke width would work around the problem in this case. The process described does not carry out any scale-dependant generalisation, and the map is built up by different users who applied different amounts of generalisation at different scales when adding the data, so it is very possible that features can be obscured by each other. Indeed, many OSM mappers may, quite correctly, not want to apply generalisation at all when recording the data, preferring to record the data itself accurately and letting downstream renders do the generalisation.
3. Hairy polylines at the edges – not a problem if a rectangular map is needed and the map extent is set correctly when preparing for printing. However it is not possible to blank out corners or pieces of the map where detail is not wanted. Instead, out-of-bounds polygons could be used.
4. Orphaned polylines at the edges – only a cosmetic problem: ways that can only be accessed by roads that are off the edge of the map may look a little odd when included on it.

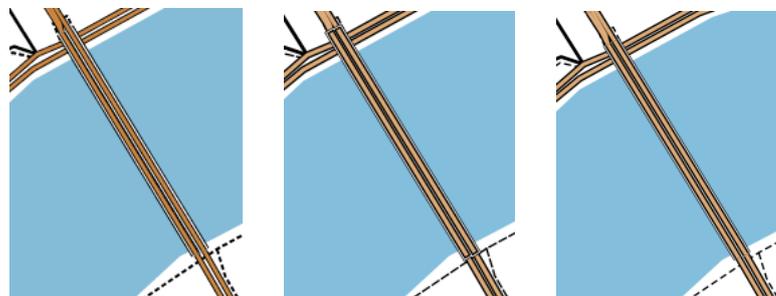


Figure 5-5: Illustrating capping in the cartography of Waterloo Bridge in London. The bridge layers are a white stroke with a thicker black stroke behind it, to create the effect of a box – this creates the correct effect of the roads crossing, not joining, at the top of the picture. **Left:** uDIG renders strokes with a flat cap, making the road appear continuous. **Middle:** Unmodified QGIS, the square cap extends beyond the bridge end, causing an apparent break in the road. **Right:** Patched QGIS, with a flat cap.

5.3.5 Data Problems

There are two main measures of the data – accuracy and completeness.

The “No-names” maps, regularly generated by the OSM project, highlight roads that do not have a name (but which would be expected to have one – i.e. not a service road). This is indicative that the road has not properly surveyed (OSM, 2008) by being visited by a contributor, but rather traced remotely from aerial images – and therefore details, such as connecting paths, may be missed. The no-names map for the Putney area is retrieved from a URL:

<http://dev.openstreetmap.org/~random/no-names/?zoom=11&lat=6717862.78346&lon=7365.80776&layers=B000>

It shows that the area has good coverage, with nearly all roads having names. There are other ways to supply names, such as using out-of-copyright maps, and even if the road was visited, the contributor may not have traced the whole length of the road, so this is only a rough guide to the accuracy of OSM coverage for an area.

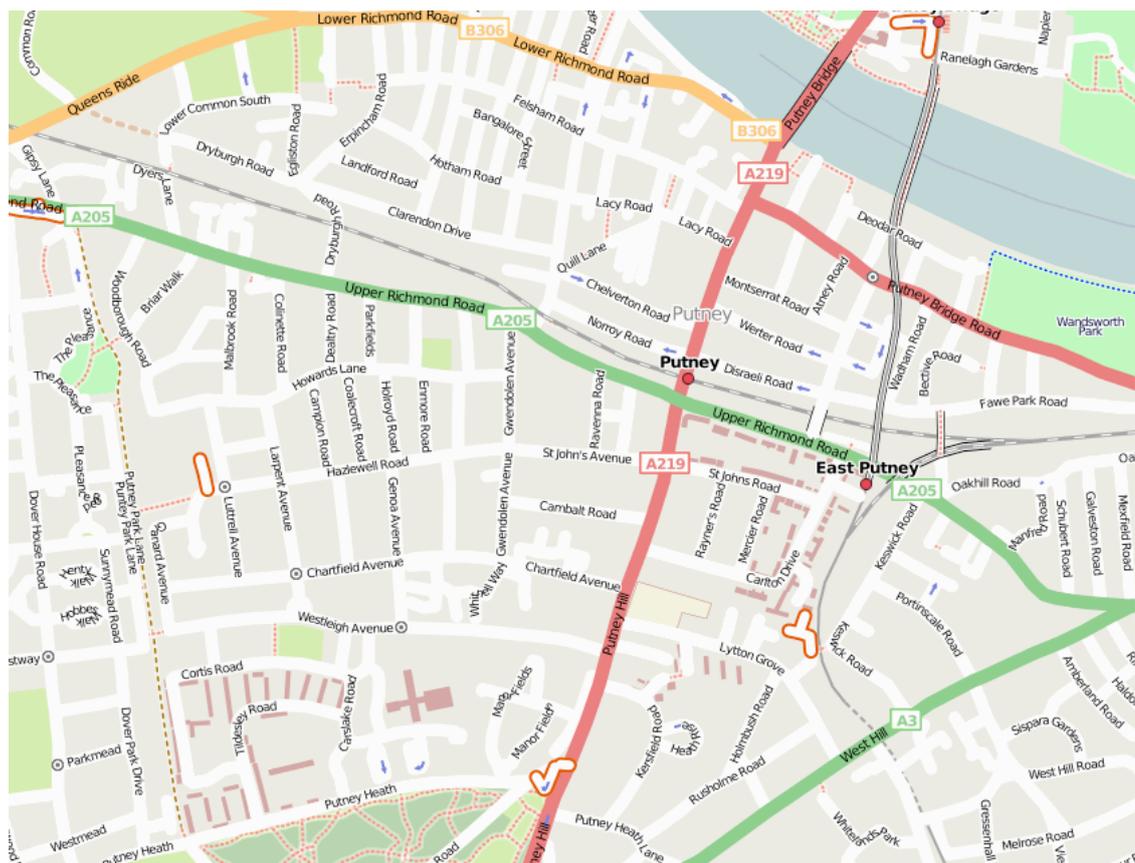


Figure 5-6: No-names map for the Putney area. Roads without names are shown with orange borders.

The completeness was discussed in the requirements analysis above – it showed that, as would be expected, a few small minor roads, mainly cul-de-sacs, are missing altogether from the old map, and a different set of roads are missing from the new map.

In terms of use for an event, the main differences are the levels of the railway lines, and a footbridge across the railway-line in the northwest, which only appears in the OSM data. The OSM data also includes paths in Putney Heath (in the south) and Bishops Park (in the north, on the north bank) – in the latter case there is almost too much detail here for this scale.

Looking at aerial photographs and other maps of the area, it looks like that some of the bridges are in fact incorrect in the old map – for example, the railway crosses over, not under, the main east-west road. However, the OSM data is also incorrect – not all of the crossings have the correct level. Looking more closely, the OSM project’s own rendered maps are wrong here too, rather than this being a problem with the level simplification carried out during the process’s import.

The places where incorrect data in the OSM base map would significantly affect the race are any missing links close to a control – such issues would be picked up in the field checking stage as each control site should still be visited, an important step in the process of creating or planning with a map for a real event. Updates based on the field check would be made in Potlatch, and the OSM data re-imported into QGIS.

5.3.6 Adding Control Data

Controls are normally shown on orienteering maps as purple circles, with the actual location of the control in the centre, and a number, or “control code”, also in purple and off to one side, where it does not obscure other features. Recently, Street-O maps have started to add a small purple (or brown) dot in the centre of the circle, to help pinpoint which side of the street the control feature is on. It is important that this dot is not obscured by other symbols, but similarly, it is important that the control circle does not obscure possible approach roads or paths to the control. Therefore, the circle should be layered underneath linear features (except streams) and the dot should be the top-most layer, or, if a single layer is to be used, transparency should be set for the circle but not the dot.

QGIS allows custom symbols to be added, in SVG format, to be added, and it should be possible to create a symbol that includes the circle, the dot, and the required transparency. However, at least on Mac OS X, this requires opening the application resource directory and placing the SVG file inside – this would need to be repeated every time the application is upgraded. For simplicity, the built in symbology, without a central dot, is used, layered behind the layers for runnable features so as not to obscure important line detail.

The orienteering layers – in this case, the “Start & Finish” and “Control” layers, are added in from the database, points can then be added to them and saved back to the database.

When retrieving the data for initial editing and future viewing, use of the “WHERE” clause in QGIS’s query builder should be made so that only features for the event concerned are retrieved – for example, entering “event_name = ‘PutneyMarch2008’” to retrieve the points that match. If subsequently adding data in to these layers, it should be noted that new points will temporarily disappear on saving unless the where clause matches the newly entered attributes, as it is re-evaluated on each save.

Layer	Feature Type	Source	Format in QGIS	Features Created by Street-O Mapper?	Intended for Final Map?
Base Map (various layers)	Polyline, Polygon	OpenStreetMap	GML files	No	Yes
Start & Finish	Point	PostGIS database	PostGIS table	Yes	Yes
Controls	Point	PostGIS database	PostGIS table	Yes	Yes

Table 5-2: Summary of the layers in the project for the Putney map.

5.3.7 Preparing for Printing

QGIS has a Print Composer function that creates a printable map out of the current view. Firstly, the page itself is configured, setting the size and orientation. Secondly, the map itself is placed on the page, scaled appropriately, and positioned so that the intended area only is visible. Thirdly, adornments, such as a title and other text, logos, a legend and a scale bar, can be dropped onto the page, and configured appropriately. Finally, the map can be printed or saved as a SVG. It should be noted that it is not currently possible to save a map for later adjustment in QGIS.

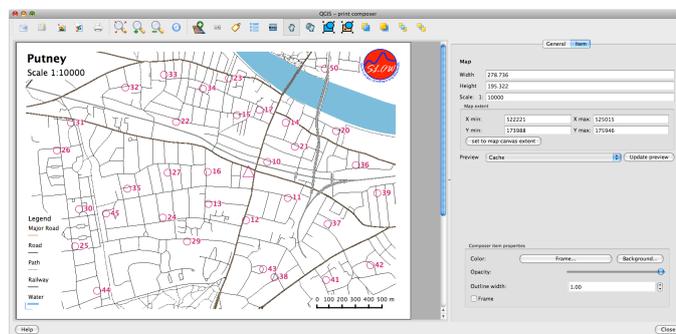


Figure 5-7: Configuring the map, on the page in the Print Composer.

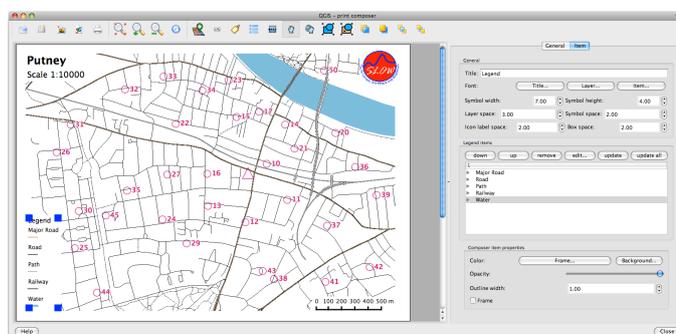
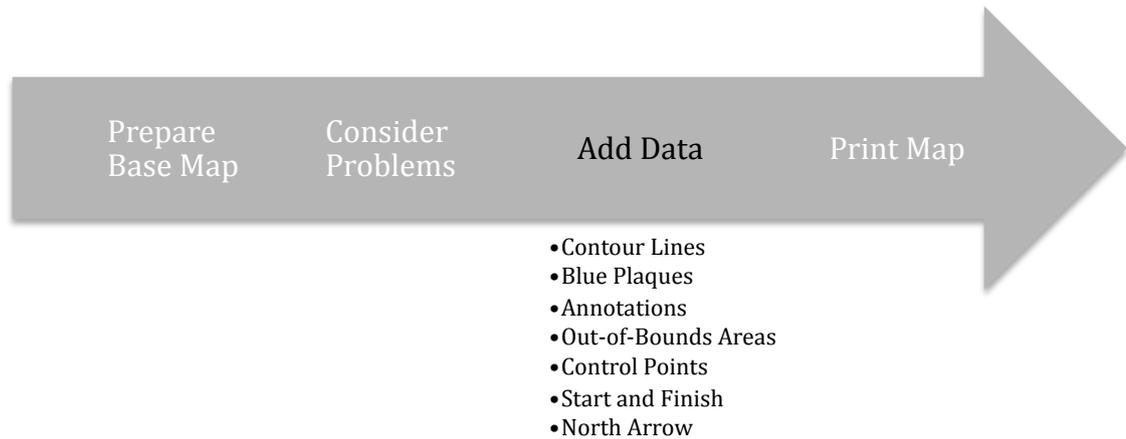


Figure 5-8: Configuring the legend adornment, which is generated from specified layers in the project.

The final map can be seen in Appendix G.

5.4 Case Study 2: Islington



This case study focuses on examples of the different types of information that can be added to the base map data to produce the final map.

The Islington area has not been mapped for Street-O events before.

First, the styled view in QGIS was created, for the relevant area, using the same technique as outlined in the first case study. The “No-names” again showed only a few roads without names. The area concerned is considerably more detailed as it is closer to central London.

The data was sourced from this URL:

<http://api.openstreetmap.org/api/0.5/map?bbox=-0.12126,51.5207,-0.08599,51.54779>

5.4.1 Adding Contours

Contours, at 5m vertical interval with 25m index lines, per the ISOM specification, were added, derived from the digital elevation model (DEM) data supplied by the NASA Shuttle Radar Topography Mission (SRTM) in 2000 and following the Shapefiles approach described by Allan (2008). The data is suitable for such fine contour detail, despite being rated with a much poorer accuracy, as the contours only need to be approximately indicative of the slope that a runner would experience rather than an accurate representation of the elevation or hill shape at any particular point.

The advantage of the technique is that it is straightforward to apply – adding contours was cited by some respondents in the questionnaire as being the most tedious part of the process, so is often not done, maps either using “uphill” arrows to show major slopes, or not showing elevation indication at all – even on steeply sloped areas such as Hampstead. The disadvantages are that the contours are jagged rather than smoothed – being derived from the DEM by an algorithm that does not generate curves. Phantom features can also

show up, as can “2D” hills that look like short lines. Both of these are artefacts of the conversion process.

In most cases, for non-flat areas, the simple addition of these rough contours is a quick and easy way to enhance the map for the competitor.

The contours are in Shapefile format, so can be edited in QGIS, to remove the artefacts, if desired. Because the contour shape-files are large and detailed, containing contour polylines that often extend far, editing can be slow in QGIS – creating a copy of the Shapefile for just the mapped area, and editing the copy, is a possible solution.

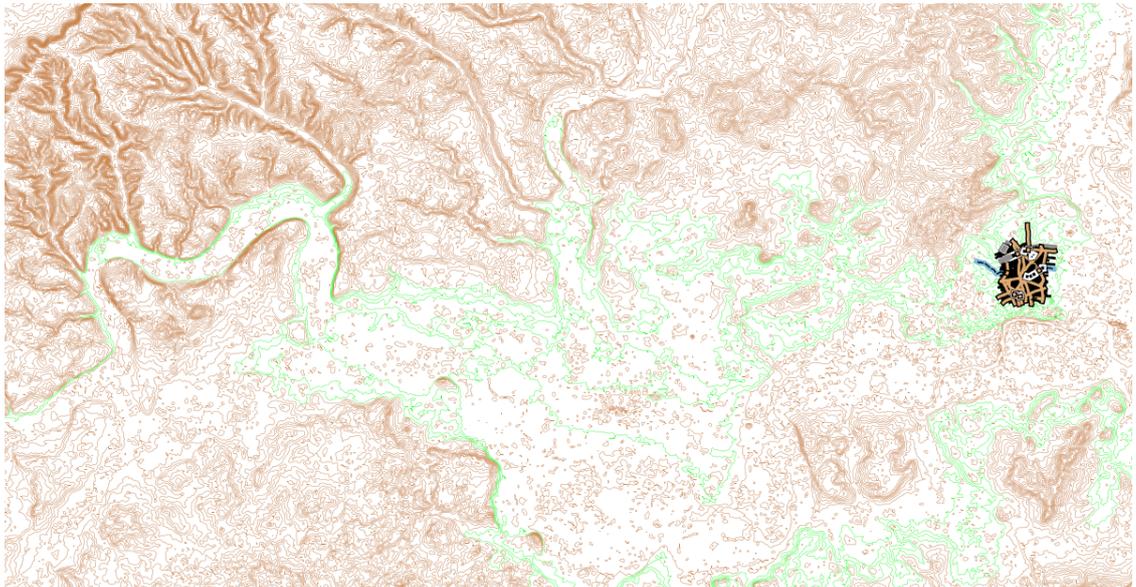


Figure 5-9: As would be expected, the contour polylines may extend far from the map – this map shows (in green) that the contours present in the mapped area extend, in some cases, well out of London. Their size and complexity means that editing them in QGIS is a slow process. The contours are derived from the SRTM DEM.

5.4.2 Adding Points of Interest

To suggest some suitable control sites for a possible event in this area, it was decided to look at some spatial datasets for interesting places in London. One of the questionnaire respondents mentioned that he uses the English Heritage Blue Plaques website (English Heritage, 2008), to find the locations of the eponymous markers on the map he is preparing. The information on each plaque is a potentially very interesting “clue” for the answer sheets that the competitors fill in during the race to prove they have been at each control.



Figure 5-10: A typical English Heritage blue plaque, this one is beside a railway bridge in East London. The inscription is “The first flying bomb on London fell here 13 June 1944”.

It was decided to look at these features as one example of using a spatial dataset unrelated to orienteering to enhance the process of creating a Street-O map for an event. Other examples, not considered in this dissertation, include spatially referenced photographs, from, for example, the Geograph or Flickr websites, both of which provide APIs that allow programmatic access to their spatial data, so can be layered in to the GIS.

Unfortunately the blue plaque data isn't freely available in a spatial format – users can search by postal district, but then are presented with a list of addresses rather than a map.

The complete set of data for the 819 plaques currently on the website was screen-scraped, converted to CSV format and tidied up. A short python program was written, to repeatedly query Google's HTTP geocoding service, part of the Google Maps API, and "geocode" – convert the addresses to a latitude and longitude. Just over 80% of the plaques successfully obtained coordinates in the London area. The program can be seen in Appendix C.

The resultant CSV file had its columns arranged and was converted to GPX format using the GPSBabel program (OSM, 2008b). The resulting file was added to the QGIS project using the GPS Tools plug-in. The controls layer from the geo-database was also added, and then chosen plaque points were added to it, as controls for the potential race.

5.4.3 Adding Orienteering Data

A layer from the geo-database – an annotations layer, was added. Points were marked on the map, to indicate areas where, from personal knowledge, a more detailed manual survey was likely to be needed. QGIS can be configured with "actions" which invoke system commands when a feature is clicked. For the annotations, a photograph could appear showing the problem.

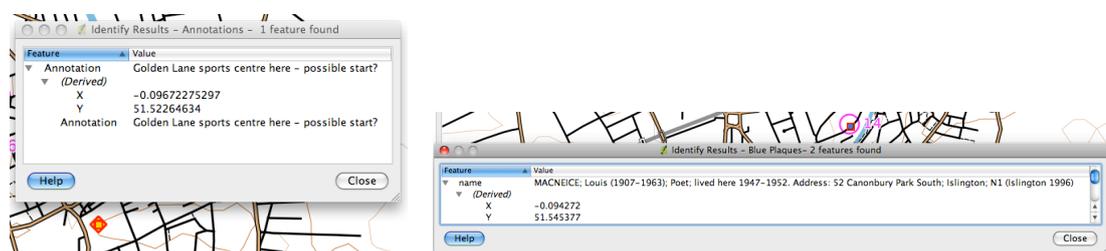


Figure 5-11: Viewing information associated with features, in QGIS, while planning a course. **Left:** Message attached to an annotation point. **Right:** Information about a blue plaque.

The intention of storing information in this way was is that future event planners for the same area, using the same geo-database, would be able to access previous annotations and if necessary act on their contents.

Three more layers from the geo-database were added, intended for features that, unlike the annotations and blue plaques, will appear on the final map – these are an areal layer

showing out-of-bounds parts of the map, a point layer for the location of the start and finish, and a point layer for the controls. All three layers were styled appropriately, using labels for the controls layer, and positioned appropriately in the list of layers. Unlike the annotations layer, the “WHERE” filter clause was used when adding the layers, ensuring that only features for the event concerned were added to the view.

Finally, a Shapefile for the north arrow was created in QGIS and added to the view – a single point was added, and styled as a north arrow. Unlike the other normal map adornments, which are added during the preparation of the map for printing, the north arrow needs to be added at this stage as a geographical feature. This is a current limitation of QGIS.

Layer	Source	Feature Type	Format in QGIS	Features Created by Street-O Mapper?	Intended for Final Map?
Base Map (various layers)	OpenStreetMap	Polyline, Polygon	GML files	No	Yes
Start & Finish	PostGIS database	Point	PostgreSQL table	Yes	Yes
Controls	PostGIS database	Point	PostgreSQL table	Yes	Yes
Out-of-Bounds	PostGIS database	Polygon	PostgreSQL table	Yes	Yes
Annotations	PostGIS database	Point	PostgreSQL table	Yes	No
North Arrow	Shapefile	Point	Shapefile	Yes	Yes
Contours (two layers)	SRTM	Polyline	Shapefiles	No	Yes
Blue Plaques	English Heritage website	Point	GPX file	No	No

Table 5-3: Summary of the layers in the project for the Islington map.

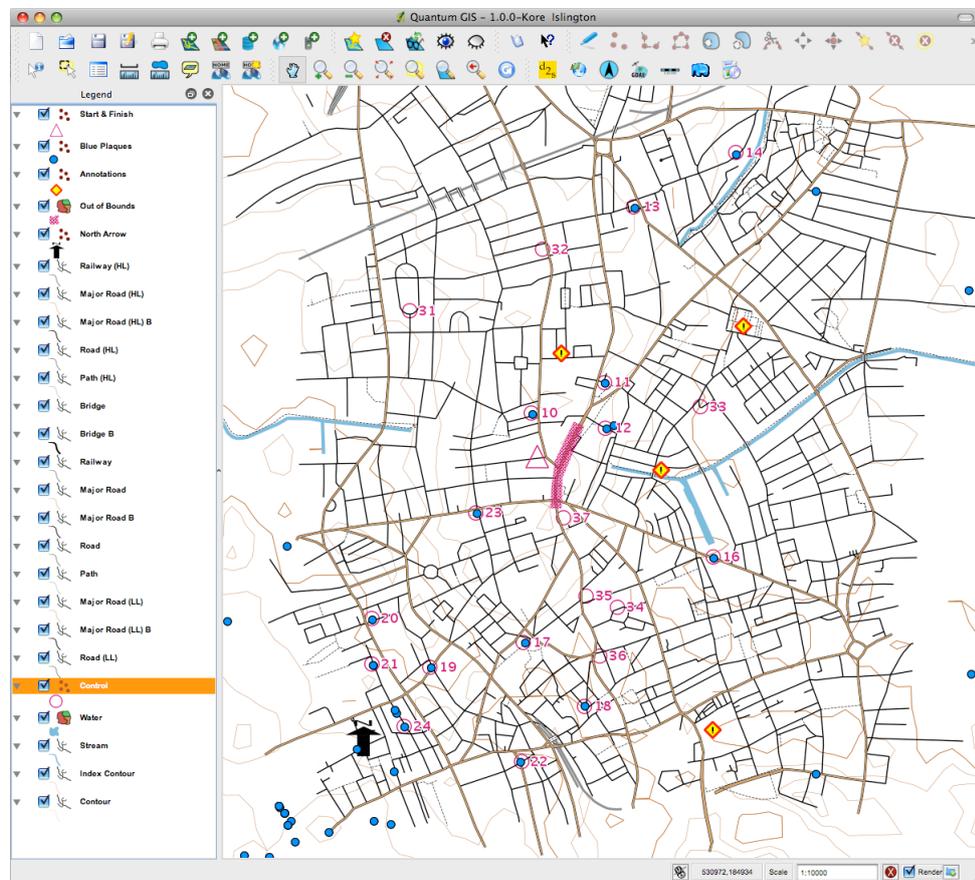


Figure 5-12: View in QGIS with the contours, blue plaque locations, controls, start/finish triangle, an out-of-bounds area, north arrow and annotation points added to a base map derived from OpenStreetMap data.

5.4.4 Producing the Map

The map was prepared for printing using the Print Composer, as described in the first case study. The map was added, positioned and scaled, and a title, legend, club logo scale-bar were added and configured in the composer. A copy of the final map can be seen in Appendix H.

5.5 Using the Data for Optimum Route Analysis

Once the base map data is retrieved from OSM, and the orienteering data is set up in the database for an event, some interesting analysis can be done with it. One such idea, which was proposed in the questionnaire and well received by many of the respondents, was being able to show the “best route” for a Street-O race, to allow competitors to compare their routes with that of an “ideal competitor”, during post-race analysis.

By using the OSM data file used as the base map, the control point locations for the event from the database, and some Python scripts written by the OSM community, it is possible to calculate, and graphically overlay, the shortest routes between two controls.

A natural extension of this would be to calculate the shortest route visiting all the controls at least once and returning to the start – the Travelling Sales Problem (TSP) (Gutin, 2002). This could then be adapted to solve the Orienteering Problem (OP) (Tsiligrades, 1984 p797), also known as the Selective TSP (Archetti, 2007), where it is not necessary to visit all the controls, each control has a varying score, and the aim is to return within a certain time, while maximising the score obtained – this is the Score Orienteering Event, the structure basis of the Street-O events in this study.

Both the TSP and OP are problems in the field of combinatorial optimization, and are notoriously hard to solve computationally, being “NP-hard” problems (Chekuri, 2008, p1), while appearing relatively easy when glancing at a map. Heuristics are often used to provide near-optimal solutions – numerous ones have been developed (Lawler, 1985 ch5-7) and the field is very active (Gutin, 2002; Chekuri, 2008; Fischetti, 1998; Tasgetiren, 2002).

It is beyond the scope of this project to try and present a solution to either problem, so this section shows how the orienteering data can be prepared, bi-control routes created, and data assembled for later analysis by a TSP solver implementation. A number of promising solvers are also looked at.

5.6 Case Study 3: Routing Between Pairs of Controls

5.6.1 Adapting Pyroutelib

The Putney map, produced in the first case study, was used to analyse routes.

Firstly, a list of control coordinates is required. This can be obtained from the geo-database with an SQL query. The PgAdmin application is a free tool for querying PostgreSQL databases, and can output query results as a CSV file.

Entering the following query produces a list of coordinates for each control.

```
select 0 as control_no, y(the_geom), x(the_geom) from "Start_&Finish" where event_name = 'PutneyMarch2008'
union
select control_no, y(the_geom), x(the_geom) from "Controls" where event_name = 'PutneyMarch2008'
order by control_no
```

Two functions supplied by the PostGIS extension, $x(\dots)$ and $y(\dots)$ were used, to access the spatial data. The union operator ensures the start/finish point is included, as well as all the controls.

An assumption was made that, because the control points weren't part of the OSM data's network, they were effectively on the road nearest to them, and any distance between them and the road was negligible. Therefore, two new roads for each point were created, linking the control to the ends of the nearest road section and so adding it to the network.

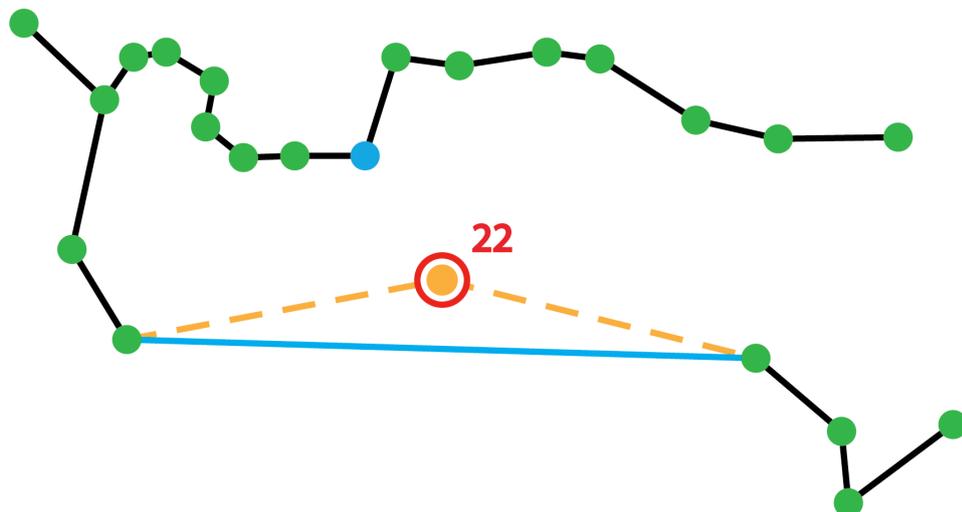


Figure 5-13: Diagram showing construction of the extra node and ways for a control. **Red Circle:** Control. **Black Lines:** Way segments. **Green Points:** Nodes. **Blue Point:** The nearest node to the control – but not the network's nearest point of approach. **Blue Line:** Nearest way segment. **Orange Dashed Line:** The two new ways added. **Orange Point:** The new node added.

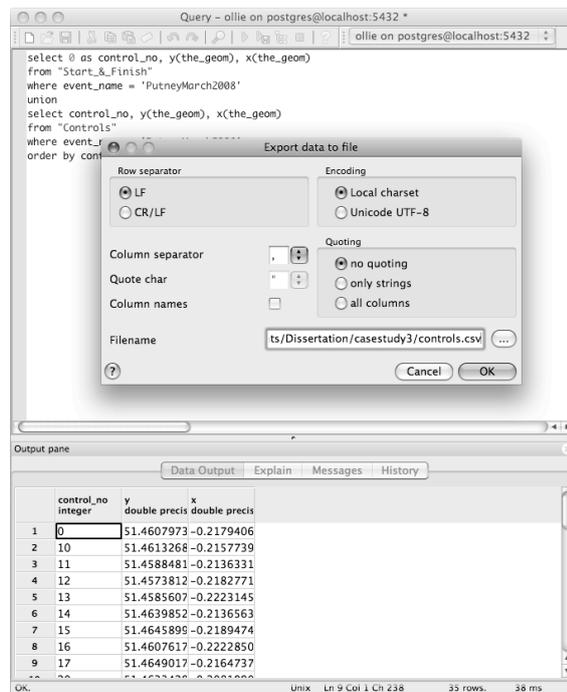


Figure 5-14: Using PgAdmin to export as CSV the locations of controls from the geo-database.

The other data source needed was the OSM XML file – the same one as used as a base map for creating the Street-O maps.

The OSM Pyroutelib python scripts (White, 2007a) were modified as follows:

- A CSV file with the control point coordinates could be input into the script, which builds up a data structure from the OSM XML data.
- The script was modified to take each control point, find the nearest point on the nearest way segment (not the same as finding the nearest point) by 2D trigonometry, and add, to the data structure, two ways leading from either end of the way segment, to the actual control site. The site itself was added as a new node, with the node ID equal to the control number.
- For simplicity, the data was not reprojected from the distance-distorting coordinate reference system associated with WGS 84, to the Cartesian British National Grid. Instead, a simple multiplication factor, approximated for the London area, was introduced to allow the latitude and longitude to be treated as x and y distances. A robust implementation of this process would need to reproject properly, using a projection that doesn't distort distances. In practice, it was found that the multiplication factor could vary considerably without affecting routing decisions, for the typical London street topology.
- Weightings were adjusted to be equal, for allowed features. This means that the shortest path is taken, regardless of whether the way is a major road, minor road or path. It is assumed that the competitive orienteer is only concerned about the

shortest route rather than a route with the best distance/aesthetic balance. The current Pyroutelib allows for various transport types, and has different weightings for each. For example, cycling routes will use quieter streets, even using a major road results in a slightly shorter journey overall.

Problems:

- Dual carriageways are modelled as two separate roads in OSM, only connecting at junctions. In practice, an orienteer would be able to cross both sections of the dual carriageway, but routing using the OSM data would only allow the crossing at a junction.
- The set of allowed features was not obtained from those used for the map creation, but rather used the existing definitions in Pyroutelib. This means that, in theory, legs could be routed across features not appearing on the map, or that the map may show allowed features apparently ignored by the routing, if there was divergence between the two sets.
- Areas where running is allowed on were not considered, such as parks – routing was treated as a strictly node-network problem. One possible solution would be to convert running areas to a number of straight lines running between every entry/exit point for the area.

Code snippets used to adapt Pyroutelib can be found in Appendix I.

The modified Pyroutelib scripts produce an OSM XML file containing the calculated best route, for each pair of controls. The OSM XML files are then run back through `osm2gml.py` to produce route polyline layer files, which can then be added back into QGIS and overlaid on the existing map.

Another script within the Pyroutelib package can generate an image, showing the features that were considered during the course of calculating the optimum route – this may be useful for “debugging” an apparently oddly chosen shortest route. The script was not modified for this project.

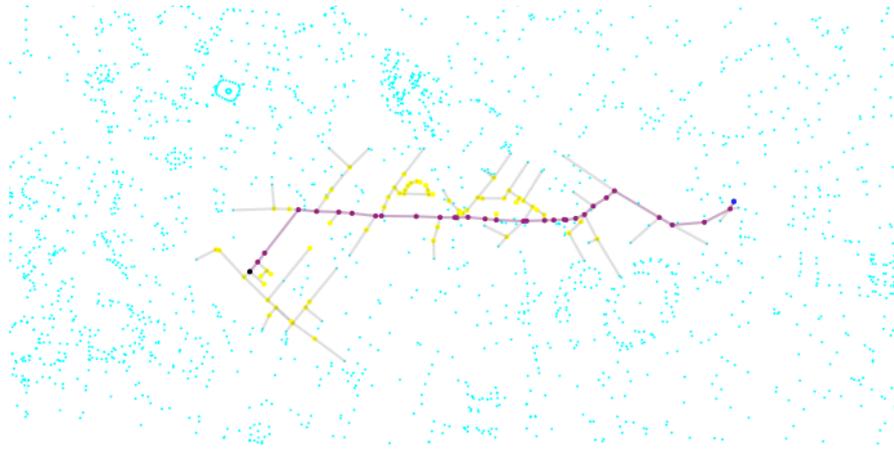


Figure 5-15: Graphic generated by `pyroute.py` (White, 2007b), showing the decision process in calculating the shortest route between two nodes. **Purple:** Nodes used. **Yellow:** Nodes considered. **Blue:** Nodes not considered. The picture has had its colours inverted from the original, for clarity.

5.6.2 Displaying Routing on the Map

The routes can then be overlaid in QGIS.

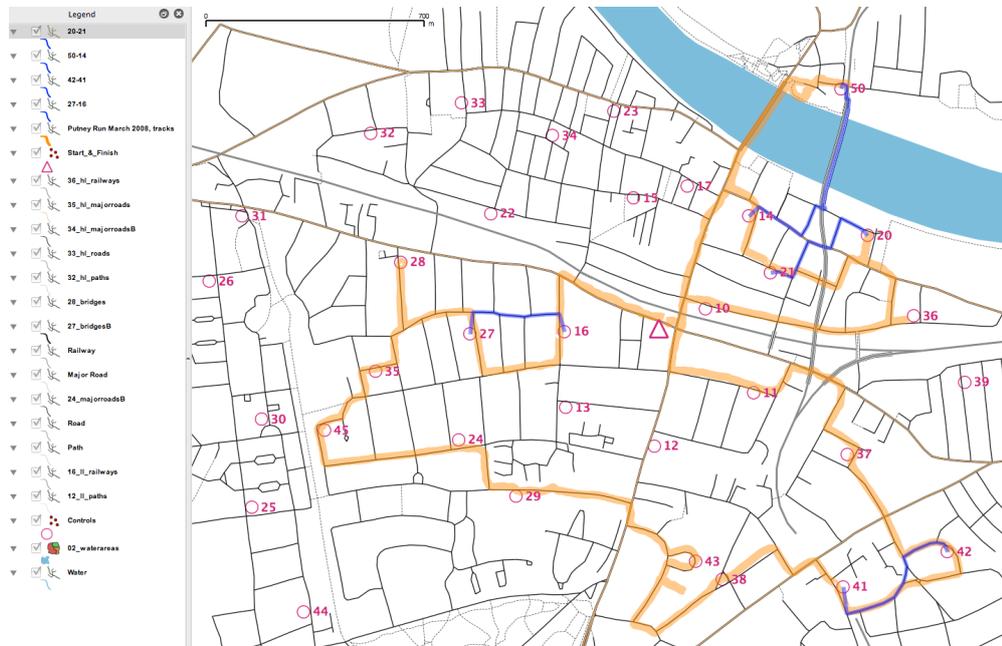


Figure 5-16: View in QGIS of the Putney map used in Case Study 1, with some routes overlaid. **Orange:** A GPS trace of the actual route run. **Blue:** Some better alternatives for legs suggested by Pyrouelib, based on OSM data.

In Figure 5-16, a number of Pyrouelib-calculated routes are displayed. In addition, a GPS track of the route taken by the author, when the Putney event was last held in March 2008, is shown. Not all controls were visited due to time constraints. Neither the base map nor the GPS data has been corrected, so, for some streets, the track doesn't line up perfectly. The calculated routes show four legs where the best route wasn't taken, for the controls visited during the course of the race.

5.7 Examples of Solving the TSP

By taking the routes generated by the modified Pyroutelib, and calculating their distances, it should be possible to assemble a matrix containing the distances between each control point and every other one. This information could then be used to solve the TSP.

A number of accessible TSP solvers are available:

- **Engdhal** (2007a) has created a “Google Maps Mashup” called Optimap. By choosing a number of points on a Google Map, an optimum route can be calculated and displayed. The TSP solving is done in JavaScript, while Google’s API is used to provide routing information between each set of two points. Routing assumes travelling in a vehicle, with speed limits and one-way street designations including in the calculation. Heuristics are used if there are more than 9 points to visit, which provide a near-optimal solution (Engdhal, 2007b). The source code is available.
- **Greg Cavill** of DR in Australia has written a solver application, ShortO, specially tailored for Street-O events. The application is used for post-race analysis at some of the Street-O events held there. Measurements of optimum link distances between likely control pairs are measured manually and supplied to the program, which uses an algorithm to work out the shortest distance between a fixed number of the available controls – outputting the shortest route of between the best-placed controls on the course, and repeating for different control quantity requirements. Control points effectively have an equal score. In effect this is a solution of the OP assuming equal prize weights. It is not unfeasible that the OSM routing technique could provide the file with the link distances, which could then be supplied to the application. The software has not been published.
- **PgRouting** is an extension to PostgreSQL that builds on PostGIS’s capabilities by providing best routes and also TSP functionality (PgRouting, 2008). Stored procedures call C++ binaries that do the routing. The major benefit of this is that, if the OSM data is added to the database and combined with the control point data, then the calculations are carried out within one system – that of the geo-database. OSM contributors have written scripts which allow easy placement of OSM XML data into a PostgreSQL database in a form which allows use of PgRouting.
- **GRASS** includes a module “v.net.salesman” TSP solver, using a heuristic. Input and output from the module is with vector maps (Blazek, 2007). As mentioned previously, QGIS does include some GRASS integration.
- **Grothmann** (1998) has written a Java applet that solves the TSP. The source code is provided, which reveals that the solver works with a matrix of bi-point distances –

the applet provides a graphical interface to build up the points, but it is feasible that the matrix could be generated by processing the OSM data and control points similar to the case study above. The applet runs very fast, using heuristics to produce an apparently near-optimal solution virtually instantly for 50 points, although repeated running gradually improves the solution.

```

Course Length Calculator
Event Location: Cathies Creek
Event date: 07 September 2008

Links: 47

Controls Distance Route
6 307.000 St-12-20-1-14-8-9-St
7 367.000 St-10-12-20-1-14-8-9-St
8 431.000 St-19-4-18-16-10-12-20-1-St
9 463.000 St-9-8-14-1-12-10-16-18-4-St
10 493.000 St-9-8-14-1-12-10-16-18-4-19-St
11 552.000 St-9-8-14-1-20-12-10-16-18-4-19-St
12 630.000 St-9-8-14-1-20-12-10-2-16-18-4-19-St
13 686.000 St-9-8-14-1-12-20-17-2-10-16-18-4-19-St
14 753.000 St-8-9-6-11-5-14-1-20-12-10-16-18-4-19-St
15 800.000 St-19-4-18-16-10-12-20-3-5-11-6-9-8-14-1-St
16 878.000 St-19-4-18-16-2-10-12-20-3-5-11-6-9-8-14-1-St
17 962.000 St-12-1-14-8-9-6-11-5-3-20-17-2-10-16-18-4-19-St
18 1043.000 St-12-20-1-14-8-9-6-11-5-3-7-17-2-10-16-18-4-19-St
19 1104.000 St-12-20-1-14-8-9-6-11-5-3-13-7-17-2-10-16-18-4-19-St
20 1191.000 St-19-4-18-16-15-13-7-17-2-10-12-20-3-5-11-6-9-8-14-1-St

```

Figure 5-17: Example output of optimum route solutions, from Greg Cavill's ShortO application. The screenshot is courtesy of the author.

None of the above implementations solve the OP – with a time limit and non-compulsory controls of varying scores. An accessible solution of the OP was not found when examining the literature.

6 Evaluation

6.1 Shortcomings of the Process

- **Curved Ways Not Shown as True Curves:** There are no curves in the maps produced using the GIS. Some of the OCAD maps did have real curves, but this type of data is not yet recorded in OSM – although there is active discussion on including it in the near future. In practice, roads with a pronounced curve are normally included with a large number of nodes and way segments to approximate a curve.
- **No Orienteering-Specific Base Map Editing:** The design choice to not allow editing of the base map in the GIS, to encourage any necessary edits to be made at source and with consideration for non-orienteering users of the data, means that some specific cartography cannot be made. For example, bridges in the OCAD maps are generally easy to spot, with exaggerated lines indicating their location, unless the area is already busy with other features. Such exaggeration and differing symbology based on other nearby features is not easily configurable in the GIS.
- **Variability in Coverage:** The difference in quality and completeness in the OSM data for different areas, due to the decentralised, volunteer and non-prescribed nature of the surveying, means that care needs to be taken when using the data – a considered visual assessment of the rendered OSM map of the area is generally always needed before it is used for a Street-O map.
- **No Selective Removal:** It is not possible to removal some instances of a feature, such as railway lines, to add clarity to a very detailed area – all or none of the features, contained in a layer, have to be present.
- **Magnetic North Projection:** There is no easy way to effectively rotate the map to align with current magnetic north – as is custom for many orienteering maps – short of creating a new projection using the PROJ4 specification. However, this issue is not a major problem due to the urban and linear nature of the running environment – compasses would normally only be used for rough direction-finding rather than fine navigation on a bearing. From personal observation, many competitors don't use a compass in Street-Os at all. Magnetic declination becomes much more important when navigating across a featureless forest, terrain not typically encountered in Street-Os.
- **Multiple Applications Needed:** OCAD has the advantage of being a single application for the surveying, tracing, drawing and production of the maps. At the minimum, the process here needs a web browser (to access the OSM data), Python

to run the conversion, and the QGIS application – although an appropriate QGIS plug-in might be able to streamline this. The process also uses a number of QGIS plug-ins with inconsistent user interface designs.

- **Application Functional Instability:** Like many open source applications, QGIS is rapidly extending and changing – there is always a danger that newer versions introduce problems or require changes in the way they are to achieve the same results, or don't provide backwards compatibility for existing maps and projects created using older versions. PostgreSQL, PostGIS and Python are less prone to this issue, as they are more mature applications.
- **API Instability** The OSM API is currently on version 0.5 – the version number indicating the beta nature of the API. A future version may change the way that downstream applications and scripts, such as the ones adapted for this project, run, or the results they produce.
- **Application Bugs and Crashes:** QGIS, in use, crashed a number of times, the crashes were generally not reproducible and happened when performing various different tasks. QGIS also has a number of bugs – the development team is generally fixing these quickly but new ones also appear. The PostGIS database also had to be reinstalled and reconfigured during the project, as the server would not start.
- **Limitations in Cartography in QGIS:** QGIS's current cartographical styles are sufficient for a simple Street-O map with the set of features shown, however more advanced cartography, such as railway lines shown with tick marks as on OCAD-produced Street-O maps, is currently not possible.
- **Differing Focus of a GIS Application:** QGIS is fundamentally built for viewing and simple analysis of GIS data – producing maps with it is a secondary function, particularly as, for professional mapping, the data is typically passed to a CAD or drawing application for finishing. It is possible therefore that the developmental focus on the application will be on improving the GIS capabilities, rather than cartographical features.
- **Technical Experience Still Needed:** Users still need to be familiar with some technical concepts, such as writing database SQL select statements including “where” clauses, dealing with projections, and understanding the distinctions of the various layer file formats and data source types.
- **Training Needed:** The process currently has numerous steps – and pitfalls, and training would be needed for non-GIS users to be comfortable with the process. It is likely however that development of an appropriate Python plug-in in QGIS, which

could drive the file format conversion, layering, styling and map layout, could automate quite a few steps.

- **Possibly Varying Results On Other Platforms:** The applications and procedures were tested on a Mac OS X computer. While it is believed that results on other operating systems, such as a Linux distribution or Windows, should be approximately the same, this may not be the case, possibly due to differing versions of some software being available for the other platforms.
- **Server Needed:** A server, specifically one that was running a recent or current version of PostgreSQL and PostGIS, or would allow installation of the software, would be needed for the process to be useable by mappers across a club or region. It may be an additional burden on a server maintainer to manage the extra software needed.

6.2 Benefits of the Process

- **Free:** The new system is completely free for the user – all the software used is available without charge.
- **No Tracing:** It eliminates what the questionnaire revealed was possibly the most tedious part of the process – tracing the base map. With a filtered set of OSM data, the base map now is the map.
- **Updates Outside Orienteering:** A worldwide community is updating the base map data, so this step doesn't need to be done by future planners in the same area. OSM is improving all the time and the quality of the data is set to continuously improve and become even more useful.
- **Benefits to OSM as well as to Orienteering:** If orienteering mappers do update the base map appropriately, using the OSM editor tools, then their work can benefit other users of the data – the benefits of utilising OSM then become two-way.
- **Centralised:** It provides for a central data repository – both OSM and a PostgreSQL server managing the orienteering data could be centralised, so the problem of finding out the person who has the most up to date map files for a particular area is mitigated. The centralised nature would also reduce unnecessary duplication of effort.
- **Standardisation:** Current Street-O maps are generally produced in version different versions of OCAD. The versions are forward compatible, but not backward compatible. This means that a person, with an older version of OCAD, would need to upgrade to a later version, to read and use a map file developed by another

mapper on the later version. With the main data being supplied by OSM and a PostgreSQL database, the issue is eliminated.

- **Fast:** A person, familiar with the process and with the technical experience, would be able to produce a Street-O orienteering map of a brand new area in under an hour, using the new process. The only additional step, which is still lengthy, would be the field-checking and control clue discovery that still requires visiting the area.

6.3 Discussion

The process outlined will be most useful for clubs that do not already have a large inventory of OCAD-produced Street-O maps, as these clubs already have the required OCAD licences and therefore would not save money by switching to this free solution.

Because of the speed at which a map can be created, once the mapper is familiar with the steps, the process is especially suited for clubs looking to start or expand a particularly active Street-O series.

The evaluation above has shown a large number of points both for and against adopting the system, and these would need to be carefully considered when deciding to set up a working real-world implementation of the system.

The implications of large-scale adoption of the design could be a significant increase in Street-O events being organised, resulting in higher visibility for the sport at its most grass-roots level.

Having more than one feasible and popular way to produce Street-O maps would be of undoubted benefit to the sport in that it would encourage innovation and development, although care would need to be taken to ensure that the data produced remains freely transferrable with any future new process. It would also provide a way for the orienteering mapping community to become more aware of GISes and what can be done with them. Further analytical uses of the data could allow creative new types of events or training techniques to be invented.

7 Conclusion

7.1 Review of Project Methodology

This dissertation reviewed existing work in the field of GIS and orienteering and determined that, while there were a number of projects that had been carried out to investigate possible uses within the sport, these were generally at a preliminary stage.

A questionnaire was conducted, targeted at people who had drawn Street-O maps recently. The intent was to understand how the current process worked, the technologies used, and the shortcomings of the system, as well as gauging awareness of GIS and opinion on some potential uses of it. The sample size was small for a questionnaire and so demographics were not extracted, however as it was carefully focused on people with the hands-on experience, it was felt that it was valid to obtain and work with the results and conclusions. The original intent was to hold interviews with the mappers, but this was deemed not necessary, as the responses received were sufficiently detailed for use in building up a requirements analysis.

A review was carried out of the various systems available for producing orienteering maps through a GIS. From this, a chosen set of systems was set up and a process designed for creating the map. This was then tested with a couple of case studies, emulating the existing cartography as much as possible. A third study used the data created in the first to demonstrate a use that could only be achieved in a GIS-based setup.

7.2 Recommendations

It is recommended that, as an initial step, PostgreSQL and PostGIS be installed on a server accessible to Street-O mappers in SLOW, and set up with a schema as outlined in this dissertation. A server owned or administered by a member of the club would be preferred for maximum flexibility, or alternatively a commercial PostgreSQL/PostGIS host could be used.

The new system should be evaluated by a mapper who has produced a number of Street-O maps using the “conventional” OCAD process, to identify any further issues, which may have been overlooked by the questionnaire respondents, but could impact the successful adoption of the new system.

Training would need to be carried out, demonstrating the process to existing mappers, who can then try creating a test map of a part of London. Prior to training, the computer system being used would need to have the needed software installed on it.

It is recommended that a trial Street-O event is carried out, using a map which is produced using the new process but where a careful field survey is also carried out, to build confidence with the new system. If an event is carried out without the careful check, then competitors should be advised before the race that the map is based on one which has not been drawn specifically for orienteering and which may be missing detail.

Care should be taken to assess the completeness and quality of OSM mapping data for areas outside of London, if creating new Street-O maps here. There are some parts of the UK where there is no OSM coverage, due to no local participation so far in the project.

7.3 Addressing the Research Question

The research question that forms the title of this dissertation asked whether a GIS and spatially aware data can add value to orienteering, in the context of producing and maintaining Street-O maps.

The answer is a “qualified yes” – it certainly can add value, but also introduces a number of complexities which need to be understood before its benefits can be fully realised. The GIS might be “two steps forward” in providing a free, open, future-proof and fast solution to creating Street-O maps, leveraging a wealth of spatially referenced general purpose data, but “one step back” in terms of simplicity and flexibility.

The real value of a GIS is one only touched on this dissertation – it is what *else* can be done with the data, once a map is created with it. The bi-control routing demonstrated, and multi-control routing proposed in the dissertation is a potentially very exciting feature which could have a real benefit in showing people how much of their event score was due to their running speed and how much was due to the good, or otherwise, route decisions they made during the race. The technique could be further developed, taking in to account conditions underfoot, weather, and gradient. Another potential is the ability so seamlessly join together maps to create more epic events. The possibilities are limited only by GIS capabilities and the ideas of the community.

7.4 Future Research Possibilities

This project has concentrated on the Street-O maps, as they are simpler than regular orienteering maps and so more suited to being produced entirely in a GIS, rather than being passed to a CAD or graphics application for the detailed cartography to be applied. As the free GIS applications evaluated in this study continue to improve and add functionality, there is potential for them to gain further cartographic functionality and so allow richer and more detailed orienteering maps to be produced solely in them.

QGIS has good integration with the Python scripting language – the whole application can be controlled using Python. It should be possible to further automate many of the steps outlined in the first two case studies above, perhaps by way of an orienteering plug-in that would lead mappers through the step-by-step process of turning the OSM data and ground survey results into a completed map, ready for printing.

There is definite potential in developing a solver, with appropriate heuristics, for the Orienteering Problem, which would accept data from OSM as outlined in the third case study. An end-to-end solution would be on laptop brought to an event, with the control points added on to the known base-map immediately after the race, and the solver then run to display the best routes people, running at various fixed speeds, should have taken.

Application of contours derived from a DEM was straightforward to do, but the contours generated during the process are not smoothed or simplified. A process that applies Bezier curves to the contours, and removes artefacts and unneeded detail, would make the contours look more “natural” and so more familiar to the competitor.

References

Note: All web references were accessed and confirmed working in September 2008.

Allan, A. et al. 2008 *Contours*, <http://wiki.openstreetmap.org/index.php/Contours>

Archetti, C. et al., 2007, *The Capacitated Team Orienteering and Profitable Tour Problems*.

Submitted version available at: <http://www.gerad.ca/~alainh/G-2007-31.pdf>. Extended

abstract available at: <http://transp-or2.epfl.ch/tristan/FullPapers/038Speranza.pdf>

BAOC 2007 *Emerald Hills*, http://www.baoc.org/wiki/Schedule/2007/Emerald_Hills

Barrable, N., 2007, *SLOW Street League*, *CompassSport* **28** (2), p18

Berlin, K., 2008, *GIS and their Usability*, <http://karsten-berlin.net/gisusability.php>

Blazek, R. & Neteler, M., 2007, *GRASS GIS: v.net.salesman*,

http://www.grass.itc.it/gdp/html_grass63/v.net.salesman.html

British Orienteering, 2008, *Mapping and Ordnance Survey*,

<http://www.britishorienteering.org.uk/event/mappers.php#Survey>

Cameron, N., 2007, *Let's Have More Street-O!*, *Orienteering Focus Autumn 2007*, p15

Chekuri, C. et al., 2008, *Improved Algorithms for Orienteering and Related Problems*,

presented at Symposium on Discrete Algorithms (SODA) 2008, Available at:

<http://www.cs.uiuc.edu/homes/chekuri/papers/orienteering2.pdf>

Dean, M., 2007, *FVO Night Street O – Dunblane*, [http://odean-](http://odean-deano.blogspot.com/2007/12/fvo-night-street-o-dunblane.html)

[deano.blogspot.com/2007/12/fvo-night-street-o-dunblane.html](http://odean-deano.blogspot.com/2007/12/fvo-night-street-o-dunblane.html)

Engdhal, G., 2007a, *TSP Solver for Google Maps*, <http://gebweb.net/optimap/>

Engdhal, G., 2007b, *Behind the Scenes of Optimap*,

<http://gebweb.net/blogpost/2007/07/05/behind-the-scenes-of-optimap/>

English Heritage, 2008, *Search Blue Plaques*, [http://www.english-](http://www.english-heritage.org.uk/server/show/nav.1499)

[heritage.org.uk/server/show/nav.1499](http://www.english-heritage.org.uk/server/show/nav.1499)

Erle, S. et al., 2006, *Converting OSM to GML*,

http://wiki.openstreetmap.org/index.php/Converting_OSM_to_GML

- Fan, C. & Zou, Z., 2005, *The Application of GIS in Orienteering*, Central South University, Changsha, Hunan, China. Masters thesis.
- Fischetti, M. et al., 1998, *Solving the Orienteering Problem through Branch-and-Cut*, *INFORMS Journal on Computing* **10** (2) pp133-148. Available at <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.2933>
- Foster, 2004, *GPX: the GPS Exchange Format*, <http://www.topografix.com/gpx.asp>
- Golicher D., 2008, *QGIS UDIG and PostGIS*, <http://duncanjg.wordpress.com/2008/03/30/qgis-udig-and-postgis/>
- Grant, D., 2008, *refNum > OpenStreetMap*, <http://www.refnum.com/osm/>
- GRASS, 2008, *GRASS: Introduction*, <http://grass.itc.it/intro/general.php>
- Grothmann, R., 1998, *Travelling Salesman*, <http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/tsp.html>
- Gutin, G., 2002, *The Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers. Ch. 1 available at: <http://www.cs.rhul.ac.uk/~gutin/paperstsp/chGYZ.pdf>
- Haklay, M. & Tobón, C., 2003, *Usability Engineering and PPGIS: Towards User Centred Approach*, *International Journal of Geographical Information Science*, **17**(6), pp577-592
- Haklay, M. and Zafiri, A., 2008a, *Usability Engineering for GIS: Learning From A Screenshot*, *The Cartographic Journal*, **45** (2) pp87-97
- Haklay, M., 2008b, *How good is OpenStreetMap information? A comparative study of OpenStreetMap and Ordnance Survey datasets for London and the rest of England, under review in Environment & Planning B and available from* http://www.ucl.ac.uk/~ucfamha/OSM%2odata%2oanalysis%20070808_web.pdf
- Halasz, S., 2005, *Interview with Steve Halasz*, http://qgis.org/index.php?option=com_content&task=view&id=52&Itemid=44
- Head, J., 2006, *Links to Professional Mappers*, <http://www.btinternet.com/~james.head/promappers.html>
- Hendricks, M. 2003 *Rapid Orienteering Map Creation with ArcPad and ArcGIS*, ESRI User Conference

Hendricks, M. 2005 *Integrating Orienteering Maps and Data into an Enterprise Geodatabase*, ESRI User Conference

IOF 2008 *Disciplines*, <http://orienteering.org/i3/index.php?/iof2006/disciplines>

Karlin, O. 2008 *Orienteering Route Choice – GIS*,
<http://www.oskarlin.com/2008/03/06/orienteering-route-choice-with-gis/>

Kirk, C. (ed.) 2004 *Evolution of Orienteering Map Making in Canada*, *Orienteering Canada* 32 (3), pp11-2

Lawler, E. L. et al., 1985, *The Travelling Sales Problem – A Guided Tour of Combinatorial Optimization*

Leung, C. 2008 *Integration of Modern GIS into Orienteering Course Planning and Map Making*, Hong Kong University Theses Online, <http://sunzi.lib.hku.hk/hkuto/record/B2977813X>

MacDonald, S., 2008, *London/Summer 2008 Mapping Party Marathon*,
<http://tinyurl.com/6989mb>

Nillumbik Emus, 2008, *Mapping Standards*
http://street.orienteering.com.au/maps/mapping_standards.html

OCAD, 2008a, *OCAD – Using GIS Data*, http://www.ocad.com/en/gis_support.htm

OCAD, 2008b, *45: Georeference a Map*, <http://www.ocad.com/howtos/45.htm>

OSM, 2008, *Greater London (what's not named) on OSM*,
<http://dev.openstreetmap.org/~random/progress/?region=london-highlightnoname>

OSM, 2008b, *Making GPX Tracks*,
http://wiki.openstreetmap.org/index.php/Making_GPX_Tracks

PgRouting, 2008, *pgRouting – Trac*, <http://pgrouting.postlbs.org/>

Ramm, F., 2008, *Shapefile download*, <http://www.nabble.com/Shapefile-download-td15411258.html>

Ryppö, J. et al., 2007, *Discussion: Street-O*,
http://www.attackpoint.org/discussionthread.jsp/message_86488

Sherman G. et al., 2007, *Quantum GIS User and Installation Guide*,
http://download.osgeo.org/qgis/doc/manual/qgis-0.9.1_user_guide_en.pdf

- Sibley, R. & Archer, D., 2007, *November Classic 2007*, *CompassSport* **28** (6), pp28-9
- Streb, K., 2008, *A History of Orienteering in Sherwood Forest*, http://www.noc-uk.org/sherwood_hist.htm
- Street C., 2001, *Welwyn Garden City Street-O*, http://www.happyherts.org.uk/pacemaker/june_2001/wgc_street_o.htm
- Tasgetiren, M., 2002, *A Genetic Algorithm with an Adaptive Penalty Function for the Orienteering Problem*, *Journal of Economic and Social Research* **4** (2), pp1-26. Available at: <http://www.fatih.edu.tr/~jesr/tasgetiren.pdf>
- Tiits, K., 2003, *Usability of Geographic Information Systems in Internet*. Masters thesis, available at http://www.hot.ee/kyllitiits/Kylli_Tiits_MSc_thesis.pdf
- Trussler, S., 2004, *Mapping using Mac computers*, *Orienteering Canada* **32** (4), p3
- Tsiligrides, T., 1984, *Heuristic Methods Applied to Orienteering*, *Journal of Operational Research Society* **35** pp797-809
- Tveite, H. et al., 2006, *International Specification for Sprint Orienteering Maps*, <http://lazarus.elte.hu/mc/specs/issom2007.pdf>
- White, O., 2006, *PDF Atlas*, http://wiki.openstreetmap.org/index.php/PDF_atlas
- White, O., 2007a, *Pyrouelib*, <http://wiki.openstreetmap.org/index.php/PyrouteLib>
- White, O., 2007b, *OSM routing, A*, cycle-filtered, python*, <http://almien.co.uk/OSM/Routing/>
- Woodcock, R., 2007, *SROC - History 1960s*, <http://www.sroc.org/menu4/his60.htm>
- Wright, P., 2008, *Street Orienteering in Derbyshire*, *Orienteering Focus* **Spring 2008**, p14
- Wunderlich A. & Hatcher R., 2008, *Rescuing Legacy Digital Data: Maps stored in Adobe Illustrator format*, *Proceedings of Digital Mapping Techniques '08*
- Kyngesburye W., 2008, *Kyng Chaos*, <http://www.kyngchaos.com/wiki/software:postgres>
- Zentai, L., 2000 *Instructor's kit for O-mapping courses*, <http://lazarus.elte.hu/mc/ik/cartogr.htm>

Appendices

A. Survey of Techniques Used by Street-O Mappers

This was conducted by email. The email was sent to club mappers in the United Kingdom, who had created or updated a Street-O map used for an event in the last couple of years. As well as answering questions relating to GIS and Street-O mapping, mappers for the London-based maps were asked to send the map files for their area. A similar email was sent to the secretary of the Nillumbik Emus Orienteering Club, based in Melbourne, Australia, which runs a very active street series in the city, and responses were received from a number of mappers in the club, and other nearby clubs, there.

Below is a copy of the version of the email sent to the UK-based mappers.

Dear orienteer,

This summer, as part of my MSc studies, I am looking at ways that the use of GIS (Geographic Information Systems) could potentially enhance the creation and maintenance of orienteering maps.

The project is concentrating on the mainly black-and-white 'Street-0' maps, as used for SLOW's Street-0 series, the SROC/SELOC Street League, and some evening score events, hosted by SAXONS, BOK and LOK last winter.

I'm contacting you as I believe you drew one of these maps or planned one of the corresponding events recently. I have a couple of requests:

FIRSTLY

I would be grateful if you could take a few moments to answer the following questions, and email me your replies. Your insight and opinions will be extremely useful for my ongoing research. Please don't feel constrained by the questions or feel you need to answer all of them.

[Q1]

Are you familiar with GIS? For example, do you work in a geographic discipline or have experience of using GIS applications such as ESRI ArcGIS or MapInfo? If so, please could you detail?

[Q2]

If you answered YES to Q1, do you have any thoughts or comments on how a GIS might be able to benefit Street-0 mapping, either creating the original map or maintaining it? And what might dissuade you from using GIS applications to make a future Street-0 map? Do you believe it would take too much time, it would be overkill for a street event, or the learning curve is too great?

[Q3]

How do you currently go about producing a Street-0 map? For example, do you use existing atlases, OS maps, mapping websites, aerial photos? Do you survey with pen and a blank sheet of paper, or make annotations on a printout? Do you go out in the streets?

[Q4]

How long does it take you to survey, and to draw, a typical Street-0 map? What parts of the process take the longest, and are most frustrating? What part of the process do you think could be made better or faster?

[Q5]

When updating a map for a forthcoming event, how do you go about this? Do you walk the whole area again? How do you choose control sites and clues? Do you do this at the time of producing the original map?

[Q6]

What software do you use to assemble and draw the final map?

[Q7]

Please let me know briefly, which if any of the following ideas could be interesting for the sport:

- i. Automatic "best route" calculation for a score event.
- ii. A system of storing potential control clues on the map itself.
- iii. A system of storing landowner information, previous problem areas and previously used sites, on the map itself.
- iv. Automated map changes based on time of year - e.g. marking a path through a path as closed if the map is printed in the winter, but open if it is printed in the summer.

[Q8]

Do you have any other comments about the process of creating a Street-0 map and planning a Street-0 event? Please let me know, any information at all may be useful.

SECONDLY

If your map is in the London area, I would be grateful if you could email me the data-file for the map. This would typically be in OCAD format, however other formats, e.g. Illustrator, would be fine.

As part of my project, I would like to, if it is OK with you, convert the London-area maps into GIS format. In doing so, they may be linked together and (for example) automated optimum route analysis could be done on them. If you would not like your map used in this way please do let me know.

Thank you very much for your help with this project, for which I would be happy to acknowledge you in the dissertation.

Yours sincerely,
Oliver O'Brien
South London Orienteers and Wayfarers (SLOW)

B. Customised Version of osm2gml.py to Create Base Map Layers

```
#!/usr/bin/python
# based on a script from http://london.freemap.in/osm2gml_simple.py.txt
# modified to work with OSM 0.5 API by Christian Zietz
# Further modified by Oliver O'Brien to produce multiple files for desired layers for Street-0 maps.
# Also converts polylines to polygons where required.

#Arguments:
#[1] OSM XML file
#[2] Type:
# allnodes - show all the nodes
# namednodes - only nodes which have a name - these are typically POIs rather than junctions.
# route - convert to GML without filtering by tags. Designed for converting line routes.
# layers - split into multiple GML files according to the tag pairs defined below - includes an overall routing network.
# Files created with no ways/nodes are deleted.

import sys, re, xml.sax, os
from xml.sax.handler import ContentHandler
from xml.etree.cElementTree import Element, SubElement, ElementTree

# tags to export if 'exportAll' is NOT set
exportTags = ["highway", "cycleway", "waterway", "natural", "railway", "layer", "bridge", "tunnel"]

# export all tags?
exportAll = 0

#TODO: Use the same filtering set that pyroute uses - currently it has a separate file (weightings.py) - combine with this to avoid duplication.
majorRoadHValues = ["motorway", "motorway_link", "trunk", "trunk_link", "primary", "primary_link", "secondary"]
minorRoadHValues = ["tertiary", "unclassified", "road", "residential", "living_street", "service", "bus_guideway"]
pathHValues = ["track", "pedestrian", "path", "cycleway", "footway", "bridleway", "byway", "steps"]
pathCValues = ["track"]
majorRiverWValues = ["riverbank", "dock"]
majorRiverNValues = ["water", "coastline", "river", "lake"]
minorRiverWValues = ["canal", "river", "stream", "drain"]
railwayRValues = ["rail", "light_rail", "subway", "tram"]
bridgeBValues = ["yes", "true"]

#QGIS loads in layers in reverse alphabetical order. So we use numbers to order the layers, to allow for easy import.
#[[layer, [[key1, [value1a, ...]], ...], include_if_in_tunnels, level, areal, ... ]
tagPairs = [
    ["34_hl_majorroadsB", [[["highway", majorRoadHValues], ["yes", "above", "no"],
    ["24_majorroadsB", [[["highway", majorRoadHValues], ["yes", "ground", "no"],
    ["14_ll_majorroadsB", [[["highway", majorRoadHValues], ["yes", "below", "no"],
    ["35_hl_majorroads", [[["highway", majorRoadHValues], ["yes", "above", "no"],
    ["25_majorroads", [[["highway", majorRoadHValues], ["yes", "ground", "no"],
    ["15_ll_majorroads", [[["highway", majorRoadHValues], ["yes", "below", "no"],
    ["33_hl_roads", [[["highway", minorRoadHValues], ["yes", "above", "no"],
    ["23_roads", [[["highway", minorRoadHValues], ["yes", "ground", "no"],
    ["13_ll_roads", [[["highway", minorRoadHValues], ["yes", "below", "no"],
    ["32_hl_paths", [[["highway", pathHValues], ["cycleway", pathCValues], ["yes", "above", "no"],
    ["22_paths", [[["highway", pathHValues], ["cycleway", pathCValues], ["yes", "ground", "no"],
    ["12_ll_paths", [[["highway", pathHValues], ["cycleway", pathCValues], ["yes", "below", "no"],
    ["36_hl_railways", [[["railway", railwayRValues], ["no", "above", "no"],
    ["26_railways", [[["railway", railwayRValues], ["no", "ground", "no"],
    ["16_ll_railways", [[["railway", railwayRValues], ["no", "below", "no"],
    ["02_waterareas", [[["waterway", majorRiverWValues], ["natural", majorRiverNValues], ["no", "all", "yes"],
    ["01_waterlines", [[["waterway", minorRiverWValues], ["no", "all", "no"],
    ["28_bridges", [[["bridge", bridgeBValues], ["no", "all", "no"],
    ["27_bridgesB", [[["bridge", bridgeBValues], ["no", "all", "no"],
    ["routingnetwork", [[["highway", majorRoadHValues], ["highway", minorRoadHValues], ["highway", pathHValues],
    ["cycleway", pathCValues], ["yes", "all", "no"],
];

counter = 0

class osm2gml (ContentHandler):
    def __init__ (self, fh, tagPair, convType):
        ContentHandler.__init__(self)
        self.fh = fh
        self.tagPair = tagPair
        self.convType = convType

    def startDocument (self):
        self.node = {}
        self.fields = {}
        self.current = None

    def startElement (self, name, attr):
        if name == 'node':
            self.node[attr["id"]] = (attr["lon"], attr["lat"])
            self.current = {'id': attr["id"]}
        elif name == 'tag' and self.current:
            k = re.sub(r'^A-Za-z0-9', '_', attr["k"])
            try:
                self.current[-1][k] = attr["v"]
            except:
                self.current[k] = attr["v"]
```

```

        self.fields[attr["k"]] = True
    elif name == 'way':
        self.current = {'id': attr["id"], 'nodes': []}
    elif name == 'nd' and self.current:
        if self.node.has_key(attr["ref"]):
            self.current['nodes'].append(self.node[attr["ref"]])

def endElement(self, name):
    global counter
    if name == 'way':
        iTunnel = self.current.has_key("tunnel") and self.current["tunnel"] not in ["no", "false"]
        tagKeyOk = 0
        tagPairOk = 0
        if self.convType == "layers":
            for tagKey in self.tagPair[1]:
                if self.current.has_key(tagKey[0]) and self.current[tagKey[0]] in tagKey[1]:
                    tagKeyOk = 1
            if self.current.has_key("bridge") and self.current["bridge"] in bridgeBValues and not
self.current.has_key("layer"):
                self.current["layer"] = "1" #force bridge sections with missing levels to be on level 1.
            if self.tagPair[2] == "yes" or iTunnel == 0:
                tagPairOk = tagPairOk + 1
            if self.tagPair[3] == "all" \
or (self.tagPair[3] == "ground" and not self.current.has_key("layer")) \
or (self.tagPair[3] == "above" and self.current.has_key("layer") and not int(self.current["layer"]) <
0) \
or (self.tagPair[3] == "below" and self.current.has_key("layer") and int(self.current["layer"]) < 0):
                #Filter out unwanted tunnels and levels.
                tagPairOk = tagPairOk + 1
            if (tagKeyOk == 1 and tagPairOk == 2) or self.convType == "route" or self.convType == "allnodes" or
self.convType == "namednodes":
                if self.convType == "layers" and self.tagPair[4] == "yes":
                    self.generateArea(self.current)
                else:
                    self.generateWay(self.current)
                self.current = None
                counter = counter + 1
    elif name == 'node':
        if self.convType == 'allnodes' or (self.convType == 'namednodes' and self.current.has_key("name")):
            self.generateNode(self.current)
            counter = counter + 1
            self.current = None

def generateArea(self, attr):
    featureMember = Element("gml:featureMember")
    feature = SubElement(featureMember, "way")
    FID = SubElement(feature, "osm_id")
    FID.text = str(attr["id"])
    geometryProperty = SubElement(feature, "gml:geometryProperty")
    polygon = SubElement(geometryProperty, "gml:Polygon")
    outerBoundaryIs = SubElement(polygon, "gml:outerBoundaryIs")
    linearRing = SubElement(outerBoundaryIs, "gml:LinearRing")
    coordinates = SubElement(linearRing, "gml:coordinates")
    coordinates.text = " ".join(map(lambda x: "%s,%s" % x, attr['nodes']))
    for k, v in attr.iteritems():
        if (k != "nodes") and (k!="") and ((k in exportTags) or exportAll):
            SubElement(feature, "" + k).text = v
    ElementTree(featureMember).write(self.fh, "utf-8")
    self.fh.write("\n")

def generateWay(self, attr):
    featureMember = Element("gml:featureMember")
    feature = SubElement(featureMember, "way")
    FID = SubElement(feature, "osm_id")
    FID.text = str(attr["id"])
    geometryProperty = SubElement(feature, "gml:geometryProperty")
    lineString = SubElement(geometryProperty, "gml:LineString")
    coordinates = SubElement(lineString, "gml:coordinates")
    coordinates.text = " ".join(map(lambda x: "%s,%s" % x, attr['nodes']))
    for k, v in attr.iteritems():
        if (k != "nodes") and (k!="") and ((k in exportTags) or exportAll):
            SubElement(feature, "" + k).text = v
    ElementTree(featureMember).write(self.fh, "utf-8")
    self.fh.write("\n")

def generateNode(self, attr):
    featureMember = Element("gml:featureMember")
    feature = SubElement(featureMember, "node")
    FID = SubElement(feature, "osm_id")
    FID.text = str(attr["id"])
    geometryProperty = SubElement(feature, "gml:geometryProperty")
    lineString = SubElement(geometryProperty, "gml:Point")
    coordinates = SubElement(lineString, "gml:coordinates")
    coordinates.text = ("%s,%s" % self.node[attr["id"]])
    for k, v in attr.iteritems():
        if (k!="") and ((k in exportTags) or exportAll):
            SubElement(feature, "" + k).text = v
    ElementTree(featureMember).write(self.fh, "utf-8")
    self.fh.write("\n")

def writeFile(fnOut, fnIn, tagPair, convType):

```

```

fileOut = open(fnOut, "w")
osmParser = osm2gml(fileOut, tagPair, convType)
fileOut.write('<?xml version="1.0" encoding="utf-8"?>\n')
fileOut.write('<gml:FeatureCollection xmlns:gml="http://www.opengis.net/gml"\n')
fileOut.write('    xmlns="http://www.openstreetmap.org/gml/">\n')
fileIn = open(fnIn, "r")
xml.sax.parse( fileIn, osmParser )
fileOut.write('</gml:FeatureCollection>\n')
fileIn.close()
fileOut.close()
global counter
print " - " + str(counter) + " items."
if counter == 0:
    os.remove(fnOut)
counter = 0

if __name__ == "__main__":
    convType = sys.argv[2]
    fnIn = sys.argv[1]
    if convType == "route":
        print "Processing route"
        fnOut = nameIn + ".gml"
        writeFile(fnOut, fnIn, None, convType)
    elif convType == "allnodes":
        fnOut = "allnodes.gml"
        writeFile(fnOut, fnIn, None, convType)
    elif convType == "namednodes":
        fnOut = "namednodes.gml"
        writeFile(fnOut, fnIn, None, convType)
    elif convType == "layers":
        for tagPair in tagPairs:
            print "Processing layer: " + tagPair[0],
            fnOut = tagPair[0] + ".gml"
            writeFile(fnOut, fnIn, tagPair, convType)

```

C. Patch Applied to QGIS to Correct Capping

The following patch was applied to the QGIS build used in this project.

```
Index: src/core/renderer/qgssinglesymbolrenderer.cpp
=====
--- src/core/renderer/qgssinglesymbolrenderer.cpp (revision 9269)
+++ src/core/renderer/qgssinglesymbolrenderer.cpp (working copy)
@@ -124,6 +124,7 @@
     {
         QPen pen = mSymbol->pen();
         pen.setWidthF( widthScale * pen.widthF() );
+    pen.setCapStyle(Qt::FlatCap);
+    p->setPen( pen );
         p->setBrush( mSymbol->brush() );
     }
 }
```

The patch improves the cartography of the bridges in the project.

For more information on capping in QT, the graphics framework used in QGIS, see:

<http://doc.trolltech.com/4.4/qpen.html#cap-style>

Information on building QGIS can be found at:

<https://trac.osgeo.org/qgis/browser/trunk/qgis/INSTALL>

D. SQL Schema for the Orienteering Tables in PostgreSQL

It is recommended that creating Shapefiles and using SPIT to set up the tables in PostgreSQL, is used, rather than using the SQL below directly.

Each table needs an entry in the `geometry_columns` table, which is part of PostGIS and specifies the geometry type, dimensionality and projection – this is updated when the tables are added with SPIT.

```
-- Table: "Controls"
-- DROP TABLE "Controls";

CREATE TABLE "Controls"
(
  gid integer NOT NULL,
  event_name character varying(80),
  control_no integer,
  score integer,
  clue_q character varying(80),
  clue_a character varying(80),
  label_x_of integer,
  label_y_of integer,
  the_geom geometry,
  CONSTRAINT "Controls_pkey" PRIMARY KEY (gid),
  CONSTRAINT enforce_dims_the_geom CHECK (ndims(the_geom) = 2),
  CONSTRAINT enforce_geotype_the_geom CHECK (geometrytype(the_geom) = 'POINT'::text OR the_geom IS NULL),
  CONSTRAINT enforce_srid_the_geom CHECK (srid(the_geom) = (-1))
)
WITH (OIDS=FALSE);
ALTER TABLE "Controls" OWNER TO public;

-- Table: "Annotations"
-- DROP TABLE "Annotations";

CREATE TABLE "Annotations"
(
  gid integer NOT NULL,
  label character varying(80),
  annotation character varying(80),
  the_geom geometry,
  CONSTRAINT "Annotations_pkey" PRIMARY KEY (gid),
  CONSTRAINT enforce_dims_the_geom CHECK (ndims(the_geom) = 2),
  CONSTRAINT enforce_geotype_the_geom CHECK (geometrytype(the_geom) = 'POINT'::text OR the_geom IS NULL),
  CONSTRAINT enforce_srid_the_geom CHECK (srid(the_geom) = (-1))
)
WITH (OIDS=FALSE);
ALTER TABLE "Annotations" OWNER TO public;

-- Table: "Out-of-Bounds"
-- DROP TABLE "Out-of-Bounds";

CREATE TABLE "Out-of-Bounds"
(
  gid integer NOT NULL,
  event_name character varying(80),
  label character varying(80),
  additional character varying(80),
  label_x_of integer,
  label_y_of integer,
  the_geom geometry,
  CONSTRAINT "Out-of-Bounds_pkey" PRIMARY KEY (gid),
  CONSTRAINT enforce_dims_the_geom CHECK (ndims(the_geom) = 2),
  CONSTRAINT enforce_geotype_the_geom CHECK (geometrytype(the_geom) = 'POLYGON'::text OR the_geom IS NULL),
  CONSTRAINT enforce_srid_the_geom CHECK (srid(the_geom) = (-1))
)
WITH (OIDS=FALSE);
ALTER TABLE "Out-of-Bounds" OWNER TO public;

-- Table: "Start_&Finish"
-- DROP TABLE "Start_&Finish";

CREATE TABLE "Start_&Finish"
```

```
(
gid integer NOT NULL,
event_name character varying(80),
the_geom geometry,
CONSTRAINT "Start_&Finish_pkey" PRIMARY KEY (gid),
CONSTRAINT enforce_dims_the_geom CHECK (ndims(the_geom) = 2),
CONSTRAINT enforce_geotype_the_geom CHECK (geometrytype(the_geom) = 'POINT'::text OR the_geom IS NULL),
CONSTRAINT enforce_srid_the_geom CHECK (srid(the_geom) = (-1))
)
WITH (OIDS=FALSE);
ALTER TABLE "Start_&Finish" OWNER TO public;
```

E. Python Script used to Geo-code the Blue Plaque Addresses

The data was scraped from the English Heritage Blue Plaques Website, then tidied up and passed through this script.

The API key has been removed for privacy reasons.

```
import csv
import urllib2

suffices = ["", "A", "B", "C", "D", "E", "F", "G", "H", "J", "K", "M", "N", "P", "R", "S", "T", "U", "V", "W", "X", "Y"]
#EH website misses letter after outcode, Google geocoder api doesn't like this, so try each manually.
#List is from wikipedia (W_postcode_area etc)

rowsf = open("gcin.csv", "rb")
rows = csv.reader(rowsf)
outf = open("gcout.csv", "wb")
out = csv.writer(outf)
rowcnt = 0
reqcnt = 0 #Google has 15,000 queries per day geocoding limit.
for row in rows:
    rowcnt = rowcnt + 1
    if row[7] == "1":
        out.writerow(row)
        continue #Already geocoded and set not to re-request.
    try:
        address = row[3].strip()
        address = address.replace(" ", "+")
        print str(rowcnt) + "(" + str(reqcnt) + ") Trying... " + address,
        for suffix in suffices:
            print suffix,
            url = 'http://maps.google.com/maps/geo?q=' + address + suffix + '&output=csv&gl=uk&key=[api key]'
            reqcnt = reqcnt + 1
            googledata = urllib2.urlopen(url)
            response = googledata.read(100)
            tokens = response.split(",")
            lat = float(tokens[2])
            lon = float(tokens[3])
            if lat > 50 and lat < 53 and lon > -1 and lon < 1: #i.e. roughly in Greater London area.
                print "OK"
                out.writerow([row[0], row[1], row[2], row[3] + suffix, row[4], lat, lon, 1])
                break
            elif suffix == "Y":
                print "Giving up - writing zeros for location."
                out.writerow([row[0], row[1], row[2], row[3], row[4], 0, 0, 0])
                break
        except IndexError:
            print "Index Error... continuing."

rowsf.close()
outf.close()
```

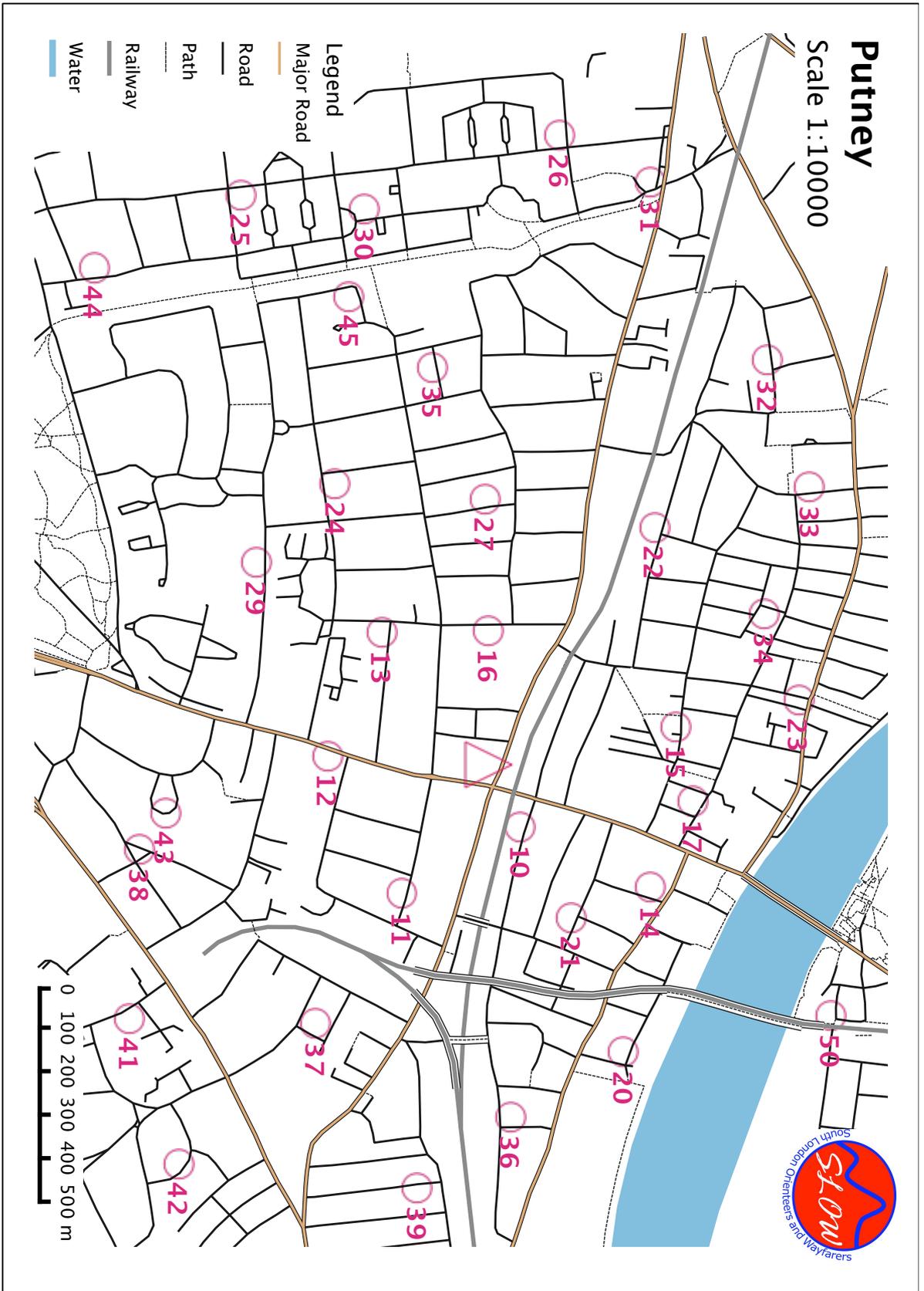
F. Original Putney Street-O Map

The map is reduced in size slightly to fit on this page.



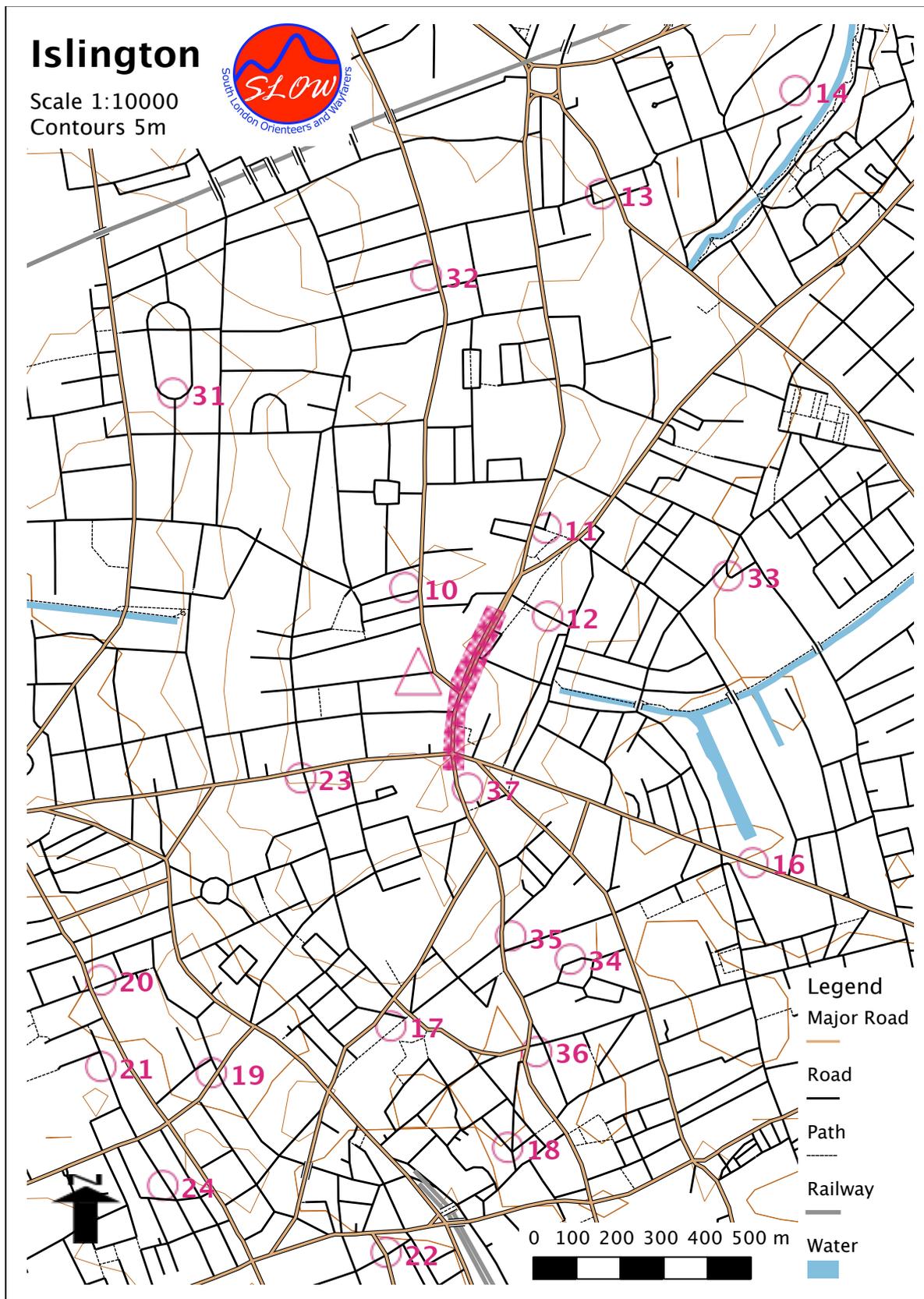
G. Putney Street-O Map

The map is reduced in size slightly to fit on this page.



H. Islington Street-O Map

The map is reduced in size slightly to fit on this page.



I. Code Fragments for Routing Street-O Controls

Some code snippets are presented here. These are not the complete set of changes to Pyroutelib that were made in the third case study.

loadosm.py

The following function was called immediately after loading the OSM XML data into the SAX Parser, to “inject” nodes for the control points from a CSV file, and ways linking them to either end of the nearest way segment:

```
def addNearNodes(self, filename):
    reader = csv.reader(open(filename, "rb"))
    for point in reader:
        bestDist = 10
        bestStartNodeId = -1
        bestEndNodeId = -1
        for currWay in self.ways:
            startNodeId = -1
            endNodeId = -1
            for currNodeId in currWay['n']:
                if startNodeId == -1:
                    startNodeId = currNodeId
                else:
                    endNodeId = startNodeId
                    startNodeId = currNodeId
            currDist = self.calcPointSegmentDistance(point, startNodeId, endNodeId)
            if currDist < bestDist:
                bestDist = currDist
                bestStartNodeId = startNodeId
                bestEndNodeId = endNodeId
        self.insertNewNode(point, bestStartNodeId, bestEndNodeId)
```

The function uses a new function, **calcPointSegmentDistance(point, startNodeId, endNodeId)**, to find the closest way segment. A similar adjustment to that in route.py was made. The trigonometry for this function was taken from a C++ function by A Partow, available at: <http://www.swissdelphicenter.ch/torry/printcode.php?id=2080>

```
def calcPointSegmentDistance(self, point, startNodeId, endNodeId):
    x1 = self.nodes[startNodeId][1]*latdegtolondeg
    y1 = self.nodes[startNodeId][0]
    x2 = self.nodes[endNodeId][1]*latdegtolondeg
    y2 = self.nodes[endNodeId][0]
    px = float(point[2])*latdegtolondeg
    py = float(point[1])
    if x1-x2 == 0 and y1-y2 == 0:
        return self.distsq(x1, y1, px, py)
    dx = x2 - x1
    dy = y2 - y1
    ratio = ((px - x1)*dx + (py - y1)*dy) / (dx * dx + dy * dy)
    if ratio < 0:
        return self.distsq(px, py, x1, y1)
    if ratio > 1:
        return self.distsq(px, py, x2, y2)
    return self.distsq(px, py, (1 - ratio) * x1 + ratio * x2, (1 - ratio) * y1 + ratio * y2)
```

A short utility function was written to avoid duplicated code above:

```
def distsq(self, x1, y1, x2, y2):
    return (x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1)
```

Finally, the new node and ways were added to the in-memory routing data structure:

```
def insertNewNode(self, point, startNodeId, endNodeId):
    #Insert the new node, at the point's lat/lon.
    self.nodes[int(point[0])] = (float(point[1]), float(point[2]))
```

```

# Store routing information
weight = getWeight('foot', 'footway')
self.addLink(startNodeId, int(point[0]), 'foot', weight)
self.addLink(int(point[0]), startNodeId, 'foot', weight)
self.addLink(endNodeId, int(point[0]), 'foot', weight)
self.addLink(int(point[0]), endNodeId, 'foot', weight)

# Store map information
if(self.storeMap):
    waynodes = []
    waynodes.append(startNodeId)
    waynodes.append(int(point[0]))
    waynodes.append(endNodeId)
    self.ways.append({ \
        't': 'footway', \
        'n': waynodes})

```

weights.py

“foot” weights were changed to 1, and also added to the “trunk” OSM tag value for the highway key. A more robust implementation would define a new user-type, “racer”, allowing the existing “foot” user-type to continue to be used for recreational routes.

route.py

The **distance(self, n1, n2)** function was adjusted, multiplying “dlon” by 1.73 to account for the different distance between longitudinal degrees and latitudinal degrees in the London area. This was a short-term workaround, as the script does not currently reproject into the distance-preserving British National Grid.

J. Dissertation Proposal

Working Title

Use of a GIS for Production and Maintenance of Orienteering Maps: Can a GIS and Spatially Aware Data Add Value to Orienteering?

Introduction

Orienteering maps are produced by both amateur orienteers, who may or may not have a cartographical or spatial background, and a small number of professionals. Production of the maps is generally done using a commercial software program, OCAD. As the dominant application in its field, it is a relatively expensive program, produces data in proprietary formats, and runs only on the Windows operating system. A number of alternative programs (Condes, Purple Pen, OORG, and OCAD's own CS edition) have started to emerge as a cheaper and more powerful way of performing one of OCAD's functions – course setting. However there is not currently a fully viable competitor for the main business of making orienteering maps. Adobe Illustrator with the free MapStudio plug-in is one alternative, but has seen only limited take-up due to the inherent costs of the free program, and does not deliver the benefits of spatially aware data.

Perhaps surprisingly, considering the many varieties and long history of GIS applications, and the large number of free solutions, there has been no significant adoption of GIS as a way of producing orienteering maps. OCAD's bespoke features for increasing the readability and "beauty" of its maps are difficult to reproduce in a data-focused GIS environment. Instead, the design-focused but spatially unaware CADs or artistic software dominates.

Aims and Objectives

This dissertation looks at ways that a GIS can potentially add value to the process of creating and maintaining orienteering maps in an amateur club environment, such as the UK orienteering club scene. The aim is to examine the state of current GIS solutions, particularly free ones, and investigate the feasibility of using and, if necessary, enhancing them for orienteering.

The aim is to harness the benefit of combining multiple maps of adjacent areas together, written by different people and clubs, and underlying them with a rich set of metadata for the process of planning and organising future events on them.

An objective is to produce a prototype-system to allow "Street-O" style maps to be produced in a GIS. These are the simplest to produce, consisting of relatively few features (generally, just control points, roads and paths) and having the potential to benefit from a systematic way of capturing current and future features useful for being a control site. Street-O events have controls that can be visited in any order, in a set time limit.

The research question can be broken down into the following sub-questions, which aim to be answered:

- Is a centralised geographic store of orienteering mapping data, and GIS usage, beneficial for organising future orienteering events, managing map updates and knowledge about an area?
- Can such a solution be delivered with free software with minimal set-up?
- Can such a solution be simple enough to be operated by non-GIS professionals?
- How does a solution make it easier to plan and organise events?
- Does an optimum-route analysis, as can be carried out once the data is in the GIS, reveal routes that orienteers did not consider, so acting as an educational tool?

Scope and Definition

The primary work will be restricted to ‘Street-O’ maps which are simpler than real orienteering maps and more suited to a GIS than the more ‘artistic’ ISOM and ISSOM based orienteering maps. The ideas and techniques will be extended to ISOM/ISSOM orienteering as a second part of the project, if time permits.

The area of study will be restricted to London and Kent, and clubs involved in the work will be restricted to LOK (North London), SLOW (South London) and SAX (Kent). I have a good working relationship with key members of these clubs and all three are known to have Street-O planners and mappers.

The technology evaluated will be restricted to free-to-use, preferably multi-platform packages that are designed for use by non-professionals and which allow easy export of their data.

Technology set up as part of the project will be as a prototype, rather than a final, production-ready product. The software will remain on my server, or my club’s server, rather than at a more generic location.

The accuracy of GPS and other modern techniques of creating the map will not be evaluated.

Research Context & Literature Review

The research context is the increasing popularity of Street-O races, particularly in London. Recently, several maps have been made, by different mappers and planners, in the two main London Clubs (SLOW and SAX). Some of these maps overlap geographically, but remain “islands” of geographic data as each map has been separately produced. This work is being undertaken to extract further value from this data by creating a “joined up” solution that is centrally managed and allows easier and faster production of maps for future races in the same format, as is likely to happen during Winter 2008-9.

The following literature is likely to be useful – the most useful items are listed first.

1. Leung, C. M. (2003) - Integration of modern GIS into orienteering course planning and map making. This is a MSc dissertation on essentially the same topic, so is very relevant. Includes a mathematical treatment of an optimum route calculation.

2. Hendricks, M. (2005) - Rapid Orienteering Map Creation with ArcPad and ArcGIS (Short online paper, presented at ESRI User Conference)
3. Hendricks, M. (2003) - Integrating Orienteering Maps and Data into an Enterprise Geodatabase (Short online paper, presented at ESRI User Conference)
4. Kerlin, O. - Orienteering route choice - GIS. <http://www.oskarlin.com/2008/03/06/orienteering-route-choice-with-gis/> (Website). Details a project involving spatial processing of orienteering-map derived data to calculate optimum routes. My dissertation may include a simpler algorithm, as there often aren't contours or terrain speed impediments in the Street-O maps.
5. QGIS Website – interview with Steve Halasz who mentions having a “future goal of using it for orienteering map production ... the plug-in system is great.”
6. Hendricks, M. – Orienteering Mapping in Maine. http://www.state.me.us/newsletter/sept2003/orienteering_mapping_in_maine.htm - this mentions an ISOM style-sheet for ArcMap.
7. Hatori, K. (2005) - GPS in orienteering mapping: Practical use of GPS in orienteering mapping for WOC2005 Japan (Presentation slides). This doesn't examine GIS, but studies effectiveness and efficiency gains of using GPS to do the fieldwork for a new orienteering map.
8. Fan, C. et al (2005) - The Application of GIS in Orienteering. http://www.paper.edu.cn/en/downloadpaper.php?serial_number=200508-40&type=1 - this short paper sounds promising but is written in Chinese.

The following are potentially useful further sources for gathering information:

- Lennon, G. – LIDAR Links for Orienteering Mapping, <http://www.lidarbasemaps.org/> - Potentially a useful resource for contacting people with GIS/orienteering experience.
- Orienteering Map Making 101 (Yahoo Group) <http://groups.yahoo.com/group/O-Map/>

Methodology

Initially, a review will be carried out of the literature, and of the GIS applications and databases suitable for use for a club managing orienteering data. This will be done primarily using the web, to ensure the information obtained is up-to-date. Planners who have produced the street maps so far will also be interviewed, to understand the current process they use to produce the maps, limitations and benefits of the process, benefits (or not) they see from a centralised, GIS perspective, and what they would like to see from such a solution.

The next stage of the project will involve implementation of a prototype system for Street-O maps. This will include deployment of a geodatabase (such as PostGIS) and a GIS application (such as QGIS). Existing data for orienteering maps will be placed in the new system, and a style-sheet will be created to allow maps to be produced from it, that essentially look the same as existing ones. If it is evaluated that creating a geodatabase for this project is too complex, then a simpler system, such as CVS for distinct map Shapefiles, may be used.

The next stage will concentrate looking at the best ways to add to the data on the geodatabase, from fieldwork and imports from other data sources. Techniques for gathering the data in the field will also be examined, including use of ArcPAD on a Trimble GeoXT, matching GPS receiver timestamps with photographs and manual notes, and Bluetooth GPS/smartphone combinations with bespoke applications for recording. A field-test on a sample area, currently unmapped for Street-O, will be carried out.

The final stage will concentrate on use of the system to create maps usable by orienteers, by creating an appropriate style-sheet, automated configuration of standard scales, boiler-plate text and so on.

The following secondary objectives will be carried out, depending on progress with the previous section:

- Optimal route calculation algorithm – detecting the optimum route to visit all controls, or the optimum route within the race time constraints for a set running speed. Easy calculation of these after an event could add real value to the competitor experience.
- Evaluation of extending the work to ISOM/ISSOM.
- Evaluation potential of the following “added value” functionality – such functionality would not be implemented during the dissertation:
 - Automated map changes based on the time of year (e.g. vegetation summer versus winter coverage may change from rough open to open. Park gates may be locked during winter or at weekends.
 - Automated water level changes – a “wet conditions” map would automatically change “intermittent stream” lines to “stream”, show pits as “wet pits” or increase the size of water features appropriately – the latter feature possibly by interpreting contours although this is likely to be unhelpful in practice.
 - Automated application of contours based on an external DEM.
 - Conversion between ISSOM, ISOM and Street-O.
 - Conversion between different scale sizes – automated generalisation and constant symbol sizes for changing scales.
 - Easy ability to join together multiple orienteering maps.
 - Automated rotation (and northing line drawing) on the map to account for magnetic-north variations.
 - Land-cover analysis of the map, e.g. 65% of map is in forest so this is a “forest” map, or there is 1.4km of disused railway line on the map.
 - Metadata for points – storage of previously used control sites, with details of what race they were used for, and planner comments about how effective they were or any problems with their use – e.g. the landowner at this point. Future possible control sites could also be stored by the mapper – visiting a site is the best way to realise its suitability for being a control site, rather than “armchair planning”. If this data was stored in the system, this could be very beneficial.
 - Metadata for areas – a layer showing each landowner’s extent on the map, with information stored on phone contacts, landowner special requests or restricted times of year for access to the land.
 - Calculation of technical difficulty of a leg, based on distance from nearest linear feature.

It is anticipated that a number of simple pieces of software will be written as part of this work.

These may include:

- Street-O style-sheet for a GIS
- ISOM or ISSOM style-sheet for a GIS (probably only as a proof-of-concept)

- Simple spatial attribute recording script for Bluetooth GPS/Symbian smartphone.
- Set up of central infrastructure for a club to manage its orienteering data produced with the new techniques. This would consist of installing and configuring the necessary software, documenting the process, and if necessary writing a simple website to manage the process. It is anticipated that most of the software will be “off the shelf” and so will not need to be written by myself.

Work Plan

w/c 23 June

- Review the literature listed above and also general Geographic Information Science literature for similar projects and useful information.

w/c 30 June

- Carry out an evaluation of GIS applications and databases suitable for this use, with mind to intended use by club planners.

w/c 7 July

- Interview club-level planners who have planned Street-O courses for their comments and ideas.
- Obtain Street-O OCAD data for Central London and Kent from SLOW, LOK and SAX, convert to a format suitable for a GIS and geo-reference them.
- Add in Meta-data based on the course controls used for Winter 2007-8.

w/c 14 July

- Set up an internet-accessible geo-database server, and add this content to it.
- Construct style-sheet for adding new Street-O data on to the mapped area.

w/c 21 July

- Evaluate and refine the techniques for adding new data and correcting old data. This includes fieldwork, and looking at automated techniques of converting the data from existing non-orienteering sources, such as OpenStreetMap’s data.
- Set up simple spatial-recording script for amateur fieldwork.
- Add my own section (London E1)

w/c 28 July

- Construct and test algorithm for optimum route calculations for Street-O races.
- Evaluate feasibility of extending the techniques to conventional ISOM/ISSOM orienteering maps.

w/c 4 August

- Unavailable (at foreign orienteering event) but may be able to further interview planners.

w/c 11 August

- Evaluate the potential of additional functionality as detailed in the Methodology section.
- Contingency week.

w/c 18 August

- Test event to try out the system and compare optimum-route calculations with real orienteers' routes.
- Complete main write-up.

w/c 25 August

- Unavailable

w/c 1 September

- Final write-up and submission.
- Contingency week.

Resources

- Field equipment
 - Symbian-series smartphone and Bluetooth GPS – supplied by myself
 - Trimble GeoXT GPS receiver and ArcPAD – will use the equipment loaned to me over the summer as part of some freelance work I am doing this summer. This will be fair use as it will not be used for additional commercial use, as explicitly forbidden in the loan agreement. It is possible however that I may lose access to this equipment after 11 August, so I will aim to have any necessary field-work using this equipment to be carried out by then.
- Club resources for staging any test Street-O events, as and when necessary. These would be supplied and insured by my orienteering club (SLOW). This includes map printing costs which would be absorbed by the club if for an event.

Ethics

The main ethical issue is ensuring people's privacy is respected when field-surveying and recording details in the field for potential Street-O control sites, and conducting any resulting events in a manner than does not impact residents' right to peaceful and private living, including clear designation of out-of-bounds areas (this is data that could be captured and stored in the GIS during the information gathering phase.)

Any request for controlled or limited use of the existing electronic data gathered by SAX, LOK and SLOW (and other clubs as appropriate) that is made when supplying the data to me, will be fully respected. Ownership of the existing data will be recording appropriately in the system.

Confidentiality

There are not expected to be any issues of confidentiality with this dissertation.

K. Reflection

This has been a most interesting project, allowing me to apply my awareness and understanding of GIS concepts and applications, acquired during the MSc teaching phase, to a sport for which I have long had a passion.

One of the most surprising aspects of the work is some of the results from the questionnaire. I never expected that the OCAD application would be so dominant in the creation of Street-O maps, mirroring its widespread use for creating more complex regular orienteering maps. I was also surprised to discover only a relatively low awareness of what GIS is and its potential, within the Street-O mapping community.

It is interesting how the process deviated from my original ideas in the proposal – initially, I was planning to use the existing Street-O vector data – but the continuous nature, independently updating and good quality of the OpenStreetMap data persuaded me that it was more suitable as a base map.

I was pleased that I was able to develop a process that did not require a significant amount of programming, either on my part or the part of the user. Utilising applications that are still being actively developed, particularly those developed with an open-source, community-led focus, can mean that programming can be required to work around bugs and rough edges.

I really feel that this work has been done at exactly the right time, for three reasons.

1. Quantum GIS, an extremely promising and user-friendly application, is virtually at the point of being released as version 1.0, having been developed for the last six years. Bug fixes and cosmetic enhancements are being checked in at an intensive rate at the moment, as preparation for the big release nears, and indeed some fairly major bugs, particularly in the map composer, which really would have stopped my whole-hearted recommendation of the process, were comprehensively fixed literally as my process was being designed.
2. OpenStreetMap, a project that began just four years ago in London, is now growing at an exponential rate. It might never be “complete”, but, in no small part thanks to a weekly series of mapping parties held around London this summer, both completeness and quality of the London data-set has improved considerably, to a point where it can be used in processes like this with reasonable confidence.
3. Finally, Street-O, as a popular informal discipline of orienteering, is really taking off in London. SLOW have seen record attendances in the 2007-8 season, and the first

event of the 2008-9 season, which happened just before submission of this work, was also popular.

It was interesting balancing the development of code and design of the process, with the documenting in the dissertation – often a great leap forward is made in only a few hours with a piece of code – but adapting the code for general use and consistent results, is the time consuming part.

I think the research question, that forms the title, was comprehensively answered, with a solution and three cases studies demonstrating of how GIS can make a real difference to managing and adding value to Street-O maps.

The dissertation has held back from the more exotic analytical capabilities of GISes and a spatially aware data-set, so there is plenty of potential for further, more interesting ideas to be developed – as is the application to other types of orienteering maps. As the cartographical capabilities of the GISes continue to grow, they will perhaps be in a position to take over more of the final map production for more and more complex maps.

I hope that this work will enable the quick creation of new Street-O areas, filling some of the gaps in London and elsewhere and leading to more events being held.