Deriving New Content from existing Cartography-Rich Map Data: the Road-to-Door Project

Sheng Zhou*, Christopher Chambers, Ben Rodgers, Jess Dyer, Elizabeth Garber

Ordnance Survey, Great Britain

January 13, 2017

Summary

This paper discusses various issues of extracting or deriving new content from existing large scale map data with substantial cartographic elements, using road to door path discovery as an example of analytical content extraction.

KEYWORDS: Derived Analytical Content, Graph, Search, Shortest Path, Ordnance Survey

1. Introduction

As the national mapping authority for Great Britain, Ordnance Survey maintains a large detailed topographic database of Great Britain with rich cartographic elements, reflecting the organisation's heritage over more than two centuries and the dominance of the Map Communication Model (MCM) since the mid-20th century (Robinson 1952; Kolácný 1969).

In recent years, geospatial applications have shifted away from the MCM paradigm, thanks to rapid advances on multiple fronts: internet, global positioning systems, mobile devices, and big data/AI/machine learning. A rise in the analytical use of spatial data has imposed a significant challenge to the usefulness (and value) of the cartography-rich data held in the OS topographic database.

The frequently quoted notion of "a map is worth a thousand words" demonstrates both the strength of cartographic maps in traditional applications as well as their deficiency in spatial analytical applications. A huge amount of information is implicitly embedded in the structure of visual elements of cartographic representations, which until recently could only be extracted by human's intelligent interpretation. By discovering such hidden information and making it explicit and machine process-able, cartographic databases will become more useful for analytical tasks. To study the feasibility of this approach, the Road-to-Door project was proposed.

2. The Road to Door Problem

Most current navigation systems rely on vehicle navigation along a pre-defined road network. A vehicle is normally guided to a road location close to the destination address. Although the path from the roadside to the destination address is usually short and straightforward, a complete set of Road-to-Door (R2D) paths was considered useful for applications such as travel-salesman-problem style route optimisations. For the small percentage of addresses where the R2D path is long or complex (as in Figure 1), path information will be very useful for users such as logistics companies, delivery drivers and the emergency services.

The term "Road to Door" is indeed better described as "Door to Road" (**find the shortest path from the entrance of a building to the nearest road) s**ince the location where a path meets a road is unknown, or more formally: in a 2D space with 1D (wall or fence) and 2D (non-accessible area)

* Sheng.Zhou@os.uk

obstacles, find the shortest path from a point on an 2D object (building) to a location on an unknown 1D element (road line) of a network in this space.

This problem can be simplified slightly by sampling the road lines to identify the shortest path between a point (on the building) and a set of points (on the road).

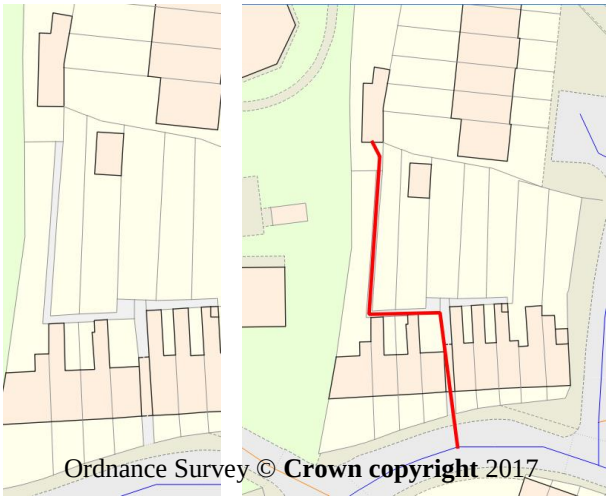

Figure 1. A complex Road-to-Door path as determined by the model we developed.

3. Road-to-Door: the prototypes

3.1 Source data

Source data used within the R2D path model included:

- Property-level addressing information from OS AddressBase Plus/Premium;
- Points sampled from OS MasterMap Integrated Transport Network (ITN) road links;
- OS MasterMap Topography layer (including topographic areas and lines, but excluding landform features) .

3.2 Process outline

In OS MasterMap (OSMM), adjacent topographic areas (TA) are separated by topographic lines (TL). An obvious solution for path finding (Figure 2) was to start from each property TA and search outwards in a **breadth-first** manner, navigating from one TA over the bordering TLs into adjacent TAs until road lines or dead ends are reached. The attributes of the two TAs and the TL separating the TAs determines the traverse-ability of the TL.

Following this idea, we re-classified TAs into 14 categories (Buildings, Gardens, Pavements, Verges, etc.) and TLs into 13 categories (Obstructing Features, Non-Obstructing Features, Roads or Track Edges, etc.) and created a 3D (14x13x14) accessibility matrix where each element denotes the accessibility for *TA_From - TL_Via - TA_To*.

Unfortunately, the attribution for TLs proved to be too simple for computing Boolean accessibility with high confidence. Some accessible features (e.g. narrow gates) are depicted as "obstructing" due to the data specification. In many of these cases, an obstructing feature such as a garden fence is shown as a solid TL whereas in reality it could have a gate or opening to enable pedestrian access to the property.

To address these issues, we used value range [0, 1] for accessibility to indicate how likely a *TA-TL-TA* configuration is accessible. For example, we give accessibility of 1.0 to *Garden - Non obstructing feature - Pavement*, and a lower value of 0.9 to *Garden - Obstructing feature - Pavement*. For path

searching, we start with an accessibility threshold of 0.95 so that only non-obstructing features are accessible; if we fail to find a path, we lower it to 0.85 to bring less likely paths into play. The threshold is stored against the final path as a confidence indicator.

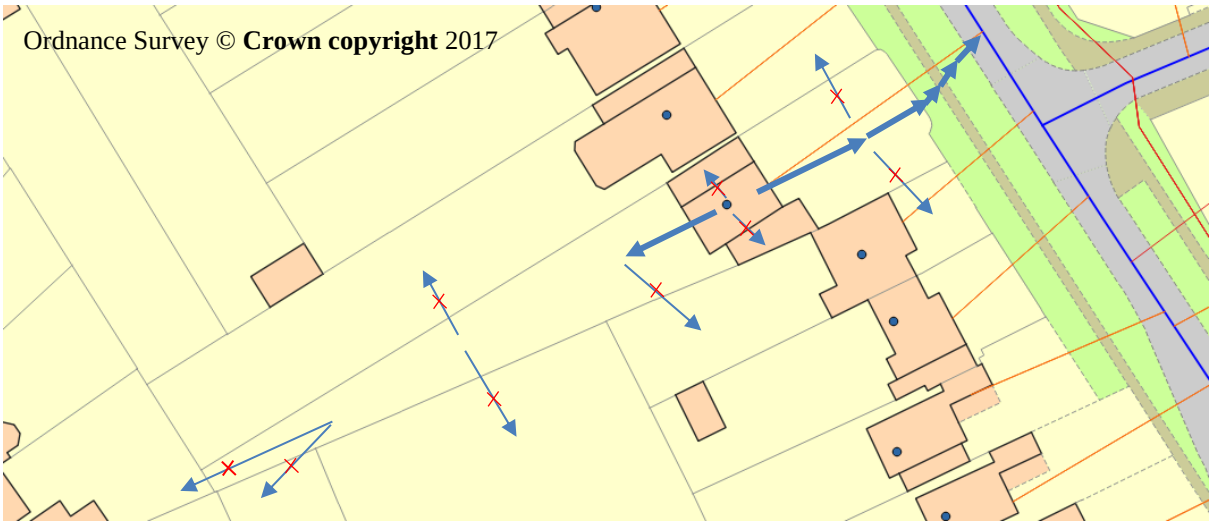

Figure 2 Map walking (red-cross denotes forbidden routes, orange line is the generated path)

In total, there are 2548 accessibility values within the matrix which allows refinements to be made in order to find a compromise between data issues and process complexity. However, it is recognised that fine-tuning one accessibility value in order to solve one problem often causes new problems in other situations. In addition, some situations required a wider context than TA-TL-TA to determine accessibility.

3.3 The data partition scheme

A partition framework was generated from ITN road links (down to minor road level) and OSMM mean high water lines. The R2D path for an address is always contained within the partition. The process runs on a per-partition basis and is therefore distributable.

3.4 Two prototypes

Our first prototype used visibility graph (VG) to find paths. Starting from the source building (Fig.3-1), all accessible TAs were searched and merged (brown area in Fig.3-2), the visibility graph (Fig.3-3) is built using Constrained Delaunay Triangulation (CDT) (Fig.3-2). Shortest paths from source to all potential destinations (points sampled from road central lines) are computed (Fig.3-4) and the shortest one is selected as the R2D path for this address (thick red line in Fig.3-5).

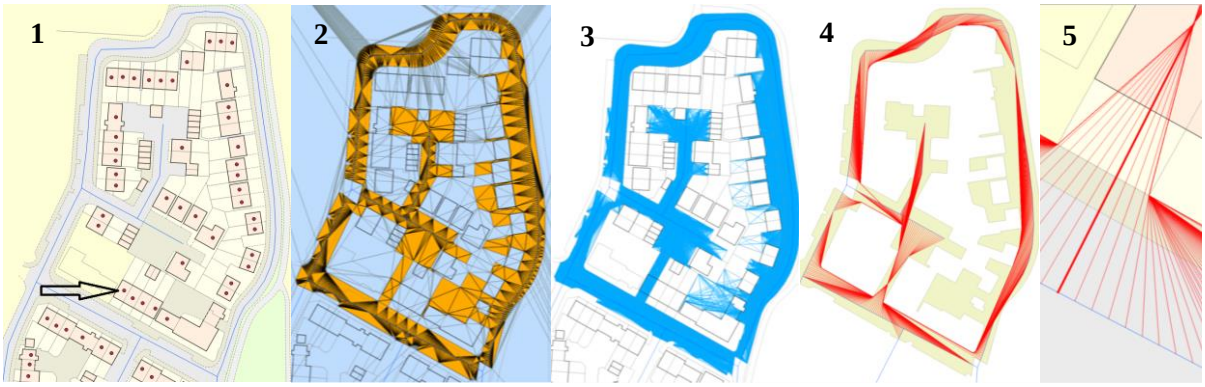

Figure 3 Finding R2D path using a visibility graph (Ordnance Survey © **Crown copyright** 2017)

The VG prototype produces near optimal results (an optimal solution was obtained with post-processing). However, VG construction is very computation intensive. It is also difficult to re-use VG for different addresses.

Our second prototype constructs a less-optimal graph from a modified definition (Figure 4) of a Triangulation Skeleton **(TS**) (Kimmel 1995). This graph may be created efficiently from the CDT of the accessible regions.

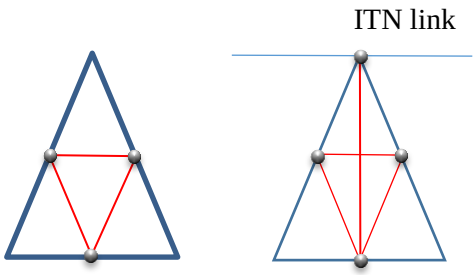

Figure 4 Graph nodes and edges (Red) from TS

Once a path is found, it may be further simplified to create a shorter path (Figure 5).

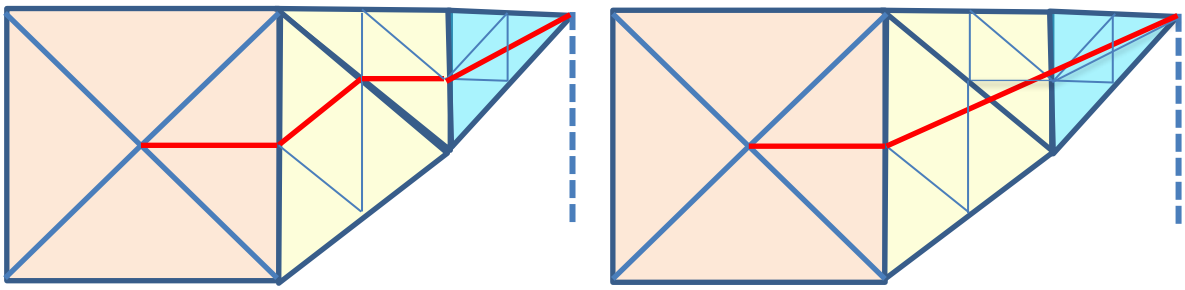

Figure 5 R2D Path from TS graph (left) and path simplification (right)

3.5 Adaptive path search

Dijkstra's algorithm (1959) was used for shortest path search with two modifications for performance improvement:

- Un-visited nodes (with infinite path length) in a prioritised queue are ordered by their geometric distance to the source
- When an R2D path is discovered (i.e. a node sampled from road is reached), its length is used to compute a threshold; un-visited nodes with distance to source greater than this threshold will be skipped; this threshold will be adjusted when a new shorter path is discovered.

4. Implementation, Initial Result and Discussion

Both prototypes are implemented in Java 8 with JTS and MGLIBJ (Java version of C++ CDT library MGLIB-2 (Zhou and Jones 2004)). Source data and results are stored in PostGIS 2.1.8. The TS-graph implementation runs more than 10 times faster than the visibility graph.

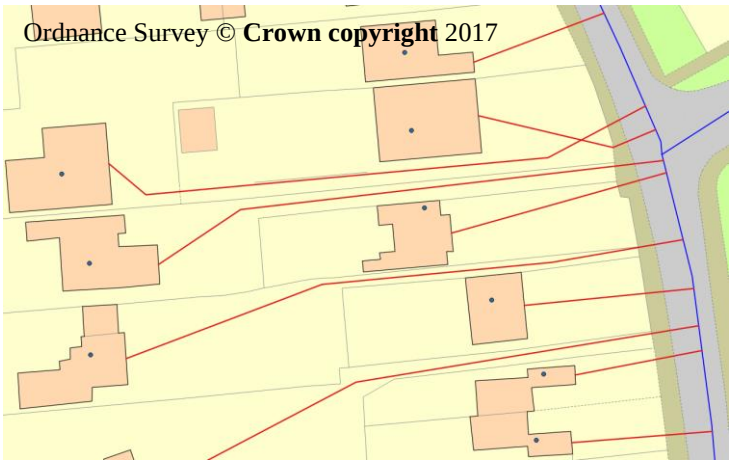

Figure 6 R2D Paths generated from TS graph

Tests were performed on three areas of over 1 million addresses in total. Initial internal evaluation has proven the prototype's potential for deriving a new data product from existing cartography-rich data.

A by-product of this process has also been the detection of anomalies within the source data. When the process fails to find a R2D path, it often signals some hard-to-detect errors within the source data (e.g. incorrectly classified TAs).

5. Acknowledgements

The authors would like to thank colleagues Simon Ware and Hazel Slawson for evaluating the prototyping results; Nick Groome and Andrew Radburn for reviewing the approach and work that was undertaken.

6. Biography

Sheng Zhou is a Senior Data Scientist at Ordnance Survey. His research interests include: application of machine learning techniques on spatial data; computational geometry; spatial databases; spatial analysis; map generalisation.

References

Dijkstra, E W (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1, 269–271.

Kimmel R, Shaked D, Kiryati N (1995). Skeletonization via Distance Maps and Level Sets. *Computer Vision and Image Understanding*, 62(3), 382-391.

Kolácný A (1969). Cartography Information: A Fundamental Concept and Term in Modern Cartography. *The Cartographic Journal*, 6(1), 47-49.

Robinson, A H (1952). *The Look at Maps: Examination of Cartographic Design*. University of Wisconsin Press, Madison.

Zhou S, Jones C B (2004). HCPO: an efficient insertion order for incremental Delaunay triangulation. *Information Processing Letters*, 93(1), 37-42