Dublin Institute of Technology
ARROW@DIT

Articles | Digital Media Centre

2010-08-24

3DQ: Threat Dome Visibility Querying on Mobile Devices

James Carswell
Dublin Institute of Technology, jcarswell@dit.ie

Keith Gardiner
Dublin Institute of Technology, keith.gardiner@dit.ie

Junjun Yin
Dublin Institute of Technology, junjun.yin@dit.ie

Follow this and additional works at: http://arrow.dit.ie/dmcart

Part of the Databases and Information Systems Commons, and the Graphics and Human Computer Interfaces Commons

Recommended Citation
Carswell, J., Gardiner, K. & Yin, J., (2010) 3DQ: Threat Dome Visibility Querying on Mobile Devices. *GIM International,* Vol.24, (8), 24, August.

This Article is brought to you for free and open access by the Digital Media Centre at ARROW@DIT. It has been accepted for inclusion in Articles by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License

3DQ : Threat Dome Visibility Querying on Mobile Devices

[Introduction]
3DQ (Three Dimensional Query) is our mobile spatial interaction (MSI) prototype for location and orientation aware mobile devices (i.e. today's sensor enabled smartphones). The prototype tailors a military style threat dome query calculation using MSI with *hidden query removal* functionality for reducing "information overload" on these off-the-shelf devices. The effect gives a more accurate and expected query result for Location-Based Services (LBS) applications by returning information on only those objects visible within a user's 3D field-of-view. Our standardised XML based request/response design enables any mobile device, regardless of operating system and/or programming language, to access the 3DQ web-service interfaces.

[First Paragraph]
A noticeable rise in the popularity of integrating sensors into state-of-the-art *smartphones* (e.g. iPhone, Android Phone, etc.) is evident in correlation with new developments in micro sensor technology. Now equipped with GPS receivers, digital compasses (magnetometers) and tilt sensors (accelerometers), today's smartphones are potentially fully position aware - providing increasingly accurate location, azimuth, and declination angles. These welcome advances dramatically promote the concept of Mobile Spatial Interaction (MSI) that can enable a mobile device to build a photo realistic and geometrically accurate cityscape model, and in turn interact with (query) the geospatial objects and attributes within the same newly created 3D scene.

Mobile Spatial Interaction
In MSI, a mobile device can act as an intermediary, linking the user to the physical environment. How to retrieve and display information about the environment intelligently is essentially at the heart of this process. With smartphones now coming pre-loaded with navigation capabilities' using web-based map services such as Google Maps, Yahoo Maps, etc., a data visualisation problem arises as the amount of information available for spatial query (e.g. geo-tagged information such as restaurants, hotels, ATMs, and other tourism/social network related POIs) and display is somewhat overwhelming for the devices and their users alike. When trying to familiarize oneself with the surrounding environment, e.g. what are these buildings around me or, is a particular point-of-interest (POI) nearby, display clutter or "information overload" becomes a significant problem. This can cause confusion and disorientation for the user, and general annoyance and apathy towards the usefulness of any LBS application. From this perspective, MSI research into the information overload problem is ongoing where map personalisation and other semantic based filtering mechanisms are essential to de-clutter and adapt the exploration of the real world to mobile devices.

Visibility Analysis
In 3DQ, we extend contemporary 2D buffer type range query functionality found in many of today's mobile LBS applications to incorporate database objects in the vertical dimension and can, for the first time on "off-the-shelf" (i.e. non-customised) mobile phones: distinguish between the various floors or objects (e.g. windows, doors, etc.) of a building while also taking into consideration the underlying DTM for determining the elevation of a user. We do this by performing directional queries on 3D spatial datasets based on a user's 3D egocentric visibility, also known as a *Threat Dome* (Figure 1).

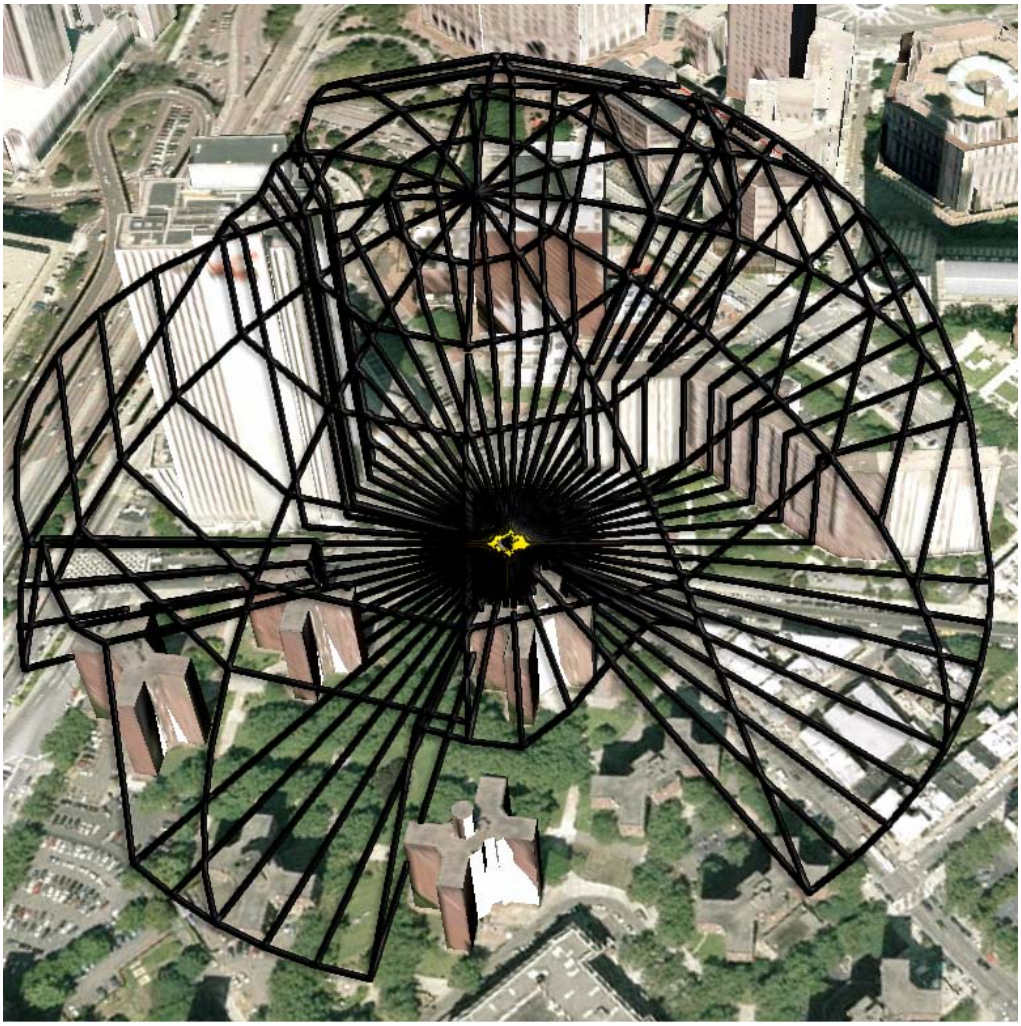

Figure 1: Threat Dome query shape interacting with 3D spatial dataset

To date, the threat dome approach to visibility analysis is typically restricted to dedicated military applications where usually a vehicle (e.g. armoured personnel carrier) wants to know in real-time what the sight lines are to/from various targets on the battlefield. In essence, a dome of a certain radius in three dimensions is generated to capture the visibility volume from the view point. The visibility volume is determined by the intersections of objects (e.g. buildings) at different elevation levels of the dome, and based on this the initial dome is generated as a view-sphere shape around the viewpoint. The position of the viewpoint is therefore visible both to and from objects that are interacting (e.g. touching, overlapping, etc.) with the dome volume - making possible the retrieval of spatially tagged information within both 2D and 3D datasets. This egocentric visibility query process enables us to progress the field of MSI research by utilizing *Hidden Query Removal* (HQR) functionality for reducing information overload. Unique to 3DQ, HQR ensures that only information on those objects that a user can actually see from their current location (and elevation), and vice-versa, get returned as results to the query. For example, new students to a university campus can explore their surroundings just by pointing their now ubiquitous smartphones at labs, offices, and classrooms to retrieve any attribute information about these objects. Spatial relationships between the 3D visibility shape and any geo-spatial objects/attributes are thus identified, answering questions such as: *"Whose office window am I pointing at?"* or; *"Are there any POIs in my field-of-view?"* or perhaps more interestingly; *"Can I see any of my Facebook/Twitter Friends from where I'm sitting?"* or indeed; *"Can they see me?"*

3DQ Components & Architecture

With large amounts of dense geospatial data needed to realize this system, a specialized Database Management System (DBMS) is required to store, index and manage these data. To perform directional queries, a database that supports 2D spatial objects is a minimum requirement. In addition, efficient topological relationship operators for determining any interactions between objects should also be available. Most of the conventional databases on the market have these capabilities, such as IBM DB2 Spatial Extender, Oracle Spatial, and PostGIS. However, performing directional queries in a 3D context becomes somewhat more complex. In addition to supporting 2D geo-spatial data management and query functions, a database that supports 3D directional queries should also be able to store and index 3D objects. For instance, a building in a 2D database is stored as polygon geometry, while in contrast it is a collection of solid geometries in a 3D database. Moreover, a 3D spatial query operator should identify not only what building is interacting with a direction vector, but also be able to distinguish which floor (or window or door, etc.) of the building it is interacting with. This advanced capability does limit the options available when choosing a database platform for this work. Taking these requirements into account, we chose Oracle 11g as the spatial database for 3DQ. In addition to supporting many coordinate systems, large sets of geo-objects (e.g. city buildings) can be stored, indexed and modelled for further query processing. 3D geospatial objects (or geometries) are modelled as composite solid, solid, surface, and polygon - corresponding to building, house, roof, and window respectively. The geometries are stored and indexed as SDO_GEOMETRY in the database, and Oracle provides the spatial operator, SDO_RELATE, to identify any 3D topological relationships. The overall design of the system, shown in Figure 2, is split into server-side and client-side respectively.

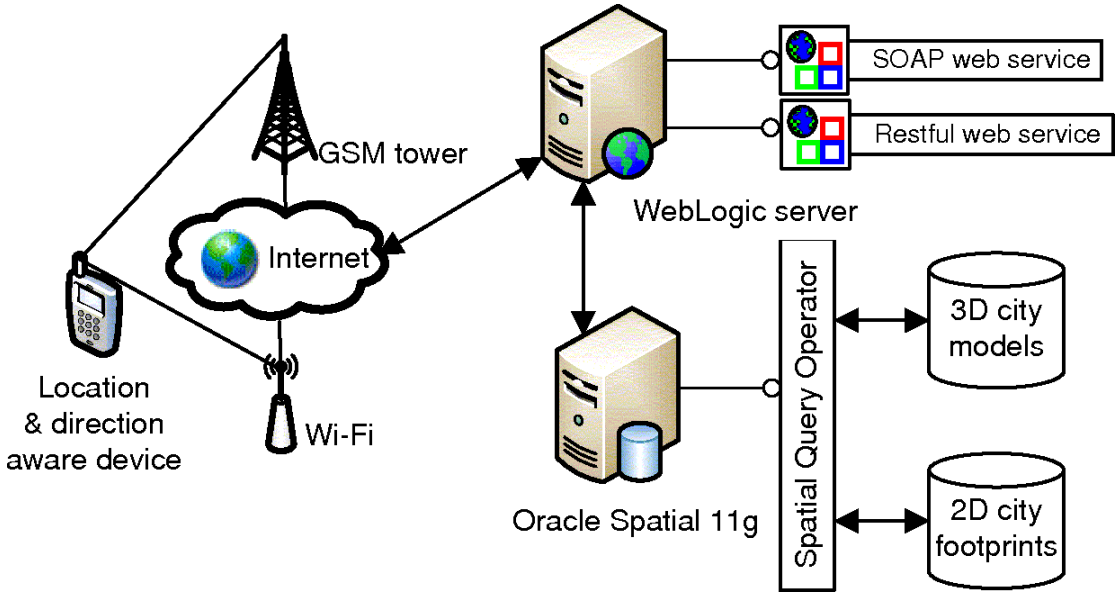

Figure 2: Overview of 3DQ Architecture

The data exchange between the client and the server is based on either a SOAP or RESTful style web-service. A WebLogic server is used for deploying the services and providing the interfaces for the mobile devices where the request from a mobile device and response returned are wrapped in an XML document – thus allowing portability across mobile platforms.

3DQ Casestudy

This section presents our prototype *Nokia Navigator 6210* smartphone with 3DQ installed and performing real-time directional queries in Dublin City (Figure 3). The

mobile device comes equipped with an integrated GPS receiver, magnetometer, and accelerometer as standard. This sensor data is accessed using Python for Symbian Series 60 (PyS60) Operating System. In order to access the web-service on the 3DQ server, a SOAP library port for PyS60 (SOAPpy) is used.

(a) (b) (c)

Figure 3: (a) 3D frustum query in real-world environment (b) Mapping interface on *Nokia Navigator 6210* with green arrow indicating query direction (c) Visualization of frustum shape interacting with 3D spatial database – note how HQR re-draws query window boundaries based on database object geometries

The entire query/retrieval process is performed on the server-side, with the primary parameters required; location, direction, and tilt provided by the mobile device. The types of 2D queries available are: standard range queries (*all neighbours* and *nearest neighbours*); single ray directional queries (*point-to-select*); and full 360º queries (*Isovist*) and directionally constrained (*Field-of-View*) queries with HQR functionality. In contrast to *Point-to-Select 2D*, and depending on the granularity of the dataset, *Point-to-Select 3D* can tell a user, for example, not only which building the device is pointing at but also which floor it is pointing at, or even the particular window it is pointing at on that floor. In 3D, the elevation of the user is taken into account along with the heights of the buildings. If a user is pointing over a lower building to a higher one behind, the 3DQ processor is capable of recognizing this difference. If a user requires detailed information about individual objects in their 3D Field-of-View (FoV), a *Frustum View* query is available to generate the required query shape and retrieve the corresponding data to the device. A frustum view query can be thought of as a “squared” flashlight beam scanning the wall of a building to get information about whatever gets “illuminated”. Finally, a *Threat Dome* query is able to provide a 360º Isovist view in three dimensions out to a specified radius. The described query processes are all implemented by generating the respective query shapes as 3D objects in a spatial database and then utilizing inherent 3D query operators to identify topological relationships.

Concluding Remarks

Further accuracy testing of 3DQ visibility queries will be carried out on a highly granular NUIM 3D campus model constructed from LiDAR data. In relation to possible alternative 3D query shapes, a new approach will investigate using FoV parameters from the phone’s on-board camera to construct a “what-you-see-is-what-you-get” shaped query frustum. This will combine reality (instead of background maps) with query results overlaid in the same display. In addition, and in keeping with current trends of publishing mobile apps on the web, we intend to investigate making available 3DQ technology to a larger audience through various online app stores (e.g. iPhone App Store, Android Market, Nokia OVI Store) for user testing and possible commercialisation opportunities.

Acknowledgements
Research presented was funded by a Strategic Research Cluster Grant (07/SRC/I1168) by Science Foundation Ireland (SFI) under the National Development Plan.

Further Reading

- Gardiner K, Yin J and Carswell JD (2009): *EgoViz - A Mobile Based Spatial Interaction System*; 9th International Symposium on Web & Wireless GIS (W^2GIS), Maynooth, Ireland: Springer LNCS Volume 5886, pp135-152.

Biography of the Author

James D. Carswell is Head of the *Spatial Information Technologies Research Group* within the Digital Media Centre of Dublin Institute of Technology. Currently he is coPI of SFI funded research into applications of context/location-aware mobile computing for environmental applications.

Keith Gardiner is a Senior DMC Researcher currently working on the SFI funded StratAG (Strategic Research in Advanced Geotechnologies) project investigating the delivery of spatial data to sensor enabled mobile devices.

Junjun Yin is a PhD student in the DMC. His research focuses on service orientated architectures for MSI. He received his BSc in Electronic Engineering from the University of Electronic Science and Technology (China) and an MSc in Geoinformatics from the University of Gävle (Sweden).

Affiliation

Dr. James D. Carswell
Spatial Scientist
Digital Media Centre
Dublin Institute of Technology
Aungier St., Dublin
Ireland
tel (+353) 1 4023264
jcarswell@dit.ie