2012-06-15

Mobile Spatial Interaction in the Future Internet of Things

James Carswell
Technological University Dublin, james.carswell@tudublin.ie

Junjun Yin
Technological University Dublin, junjun.yin@tudublin.ie

2012-06-15

Mobile Spatial Interaction in the Future Internet of Things

James Carswell

Mobile Spatial Interaction in the Future Internet of Things

*James D. Carswell and Junjun Yin
Digital Media Centre, Dublin Institute of Technology
Aungier St. Campus, Dublin, Ireland
*Corresponding author, e-mail: jcarswell@dit.ie

***Abstract*—Research and development of mobile information systems in the *Future Internet of Things* is about delivering technologies built around management and access to real-time heterogeneous datasets. Analyzing these enormous volumes of disparate data on mobile devices requires context-aware smart applications and services. 3DQ (Three Dimensional Query) is our novel mobile spatial interaction (MSI) prototype for data mining and analysis on today's location and orientation aware "smartphones" within such 3D sensor web environments. Our application tailors a military style *threat dome* query calculation using MSI with "hidden query removal" functionality to reduce information overload and heighten situation awareness on these commercial off-the-shelf (COTS) devices. Allied 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 the processing/display limitations of mobile devices. We propose that another way to filter this information is to intelligently refine the search space. The combined effect gives a more accurate and expected query (search) result for Location-Based Services (LBS) applications by returning information on only those objects/sensor enabled "things" visible within a user's 3D field-of-view (FOV) as they move through a built environment.**

Keywords- Mobile Spatial Interaction; Geoprocessing Web; Isovist; Threat Dome

I. INTRODUCTION

Geospatial information is increasingly recognized as the common denominator in both today's "web 2.0" peer-to-peer social network era and tomorrow's "web 4.0" – where it is envisioned that the Internet becomes connected to trillions of micro-sensors placed into real-world objects of all types (i.e. mechanical and non-mechanical), all with their own 128 bit IP address [1]. In other words, an *Internet of Things* that collects and sends time-stamped data to the cloud every second about their location, movement, plus any number of other measureable phenomena – e.g. environmental data such as air/water quality, ambient light/noise/radiation data, energy consumption, etc.

Some key research issues in this web 4.0 era requires new technology for enabling citizens to mine, analyse, and even affect this massive web of interoperable objects – for example by remotely turning IP connected objects such as light switches or other power consuming devices on/off. The application described in this paper proposes a **mobile sensor web data mining tool** that supports efficient capture and convenient analysis of geospatial and sensor web datasets already available today with a view to easily include *Future Internet* data sources as they come online.

Today, most new smartphones (e.g. iPhone, Android Phone, etc.) are equipped as standard with GPS receivers, digital compasses (magnetometers) and tilt sensors (accelerometers/gyros). Also, they come pre-loaded with navigation capabilities using web-based map services such as Google Maps, Yahoo Maps, or Bing Maps among others. However, even now a data visualisation problem arises as the amount of information available for spatial query and display – which includes traditional built environment objects (e.g. buildings, roads, etc.), contemporary geo-tagged information (e.g. tourism or social network related POIs), and objects from the sensor web (e.g. environmental sensors monitoring weather, air pollution, noise, etc.) - is somewhat overwhelming for these devices and their users alike. For example, when trying to familiarize oneself in an unknown environment - e.g. "What are these buildings around me?" or, "What points-of-interest (POIs) are nearby?" or, "What are the air pollution levels along this street?" - display clutter or information overload can become a significant problem. This can cause confusion and disorientation for the user, and general annoyance and apathy towards the usefulness of any Location-Based Service (LBS) application.

In MSI research, how to address the problem of optimizing information to personal needs through intelligent retrieval and display is essentially at the heart of this process. Previous work in this regard proposed a Local Visibility Model which searches for information by considering a user's actual field-of-view [2]. In [3], a 2D directional-query processor is built on a similar visibility model, where building block outlines are treated as "footprints" used to geographically clip the visible search space in the horizontal plane. Other approaches for reducing information overload look at using map personalisation techniques that monitor past user task/map behaviour to assist and/or recommend next levels of display detail [4,5,6,7]. Semantic based filtering mechanisms are designed to exploit ontologies and RDF (Resource Description Framework) descriptions of linked data and networked knowledge to essentially de-clutter and adapt the exploration of the real world to the processing/display limitations of mobile devices [8]. In our era of ever increasing multi-sensored environments, we propose that another way to improve contextual relevance for users is to spatially filter this mounting

Research presented in this paper was funded by a Strategic Research Cluster Grant (07/SRC/I1168) by Science Foundation Ireland (SFI) under the National Development Plan. The authors gratefully acknowledge this support..

data store by intelligently refining the search space in 3 dimensions.

To date, the *threat dome* approach to visibility analysis is typically restricted to the battlefield in 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 target positions. In **3DQ** (Three Dimensional Query) we extend contemporary 2D buffer type range query functionality found in many of today's mobile LBS applications to now incorporate database objects and spaces in the vertical dimension. More specifically, we move from traditional GIS desktop processing environments to a threat dome visibility query paradigm tailored to function in urban environments on smartphones. The result is a more accurate and expected search result for mobile LBS applications by returning information on only those objects visible within a user's 3D field-of-view (Fig. 1).

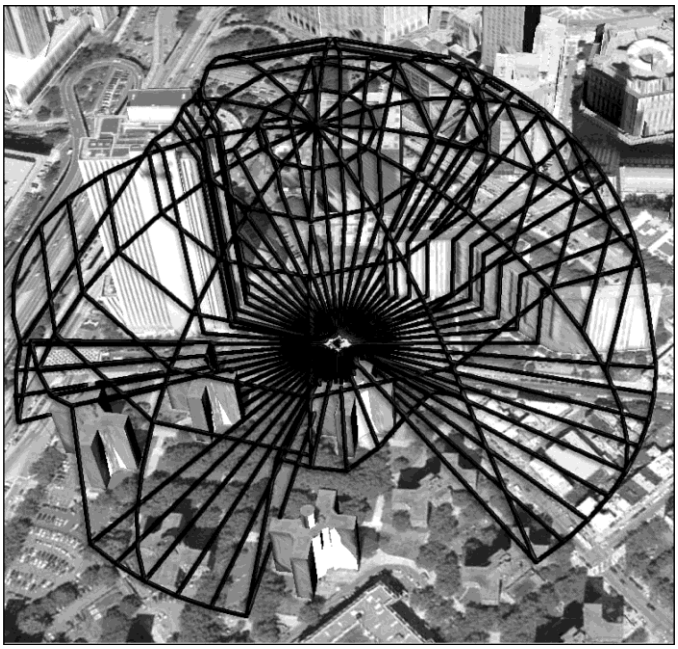


Figure 1. Threat Dome search space interacting with 3D cityscape model; only information intersecting the 3D dome volume gets returned by the query

This effectively applies *hidden query removal* (HQR) functionality in 360º 3D that takes into account both the elevation of the user and the 3D cityscape when calculating the 3D search space (volume), significantly reducing display clutter and information overload on these COTS (commercial off-the-shelf) devices. Unique to 3DQ, HQR ensures that only information on those objects (e.g. buildings, floors, windows, doors, street furniture, environmental sensors, etc.) that a user can actually see from their current location (and elevation), and vice-versa, get returned as search results to the mobile device. For example, visitors can now explore both their horizontal and vertical surroundings by pointing their smartphones at stores, offices, monuments, or any space to retrieve from the web any recorded information about these objects - answering specific questions such as: "Whose office window is that up there?" or more generally; "What are the UV readings on this beach?" or perhaps more interestingly; "Can I see any CCTV cameras from where I'm sitting?" or indeed; "Are they seeing me?". Conversely, subtracting threat dome query results from an all encompassing 2D range query of the same space will retrieve only those objects that are just out of sight. Analogous to scenes from *Star-Trek*, with Spock scanning his "tri-corder" for readings of life and atmospheric conditions on some strange world, such situation awareness capabilities would also be very interesting to bikers, joggers, walkers, city workers, and all concerned parents and citizens alike on this world who want to know, for example, the health of their immediate environment at any point in time.

First steps towards enabling this "Big Data" future are beginning with some cities, e.g. Toronto, Vancouver, Ottawa, Edmonton [www.toronto.ca/open/], and indeed countries, e.g. U.K., U.S.A. [http://data.gov.uk/, www.data.gov/catalog/], now compiling and opening data catalogues of heterogeneous spatial and non-spatial datasets to citizens for royalty-free use world-wide [9,10]. In Ireland, such open access to data is currently under discussion but not as yet freely available. Therefore, our initial 3DQ test beds include the NUIM university campus overlaid with weather sensors and parts of Dublin City overlaid with a sparse noise and air pollution sensor network.

II. 3DQ ARCHITECTURE

Due to the limitations of batteries, CPU capabilities, and available onboard spatial databases for mobile devices, the sw architecture for mobile query systems has evolved to the point where the query processes are necessarily performed server-side. Also, operational difficulties can occur as the diversity of operating systems that run on various mobile devices limit the access to their integrated sensors (e.g. GPS, accelerometers, compass, etc.), which are themselves programming language specific. This requires the server to provide different interfaces to communicate with each mobile programming language. It is due to the inherent complexities of communicating with often proprietary mobile sw/hw architectures that much current research has been focused on addressing this issue. Significantly, a promising approach built on web-services was proposed that attempts to overcome these problems by providing a standardized request/response interface between the client and the server [11,12,13].

To provide *Connect and Play* services to the majority of location and orientation aware smartphones available today, regardless of their operating systems (e.g. iPhone OS, Nokia Symbian/Windows Phone7, Google Android), 3DQ implements a robust and scalable web-service orientated interface on the server-side that employs two types of web services to ensure that 3DQ interfaces are open and universal: *Simple Object Access Protocol* (SOAP) and *Representational State Transfer* (RESTful). In this case, the exchange data format for SOAP is XML and RESTful provides output in both XML and JSON (JavaScript Object Notation). Both XML and JSON are text formats that are completely operating system language neutral. With these services, any mobile device is now free to choose a suitable programming language (e.g. Objective-C for iPhone, Java for Android, C# for Windows Phone7) based on the needs of the client-side applications, usability, and operating restrictions (e.g. getting data from the

onboard sensors), and not on the communication restrictions between the client and server.

A. 3DQ Client-Side Design

To interact with the application, a simple user interface was developed to manage query requests and display any results returned. To perform visibility based directional queries, the primary requirement is the user's current location and pointing gesture data (i.e. x,y,z, azimuth, and declination of the device). The parameters are read and recorded from the integrated sensors on the mobile device and sent to the server. Specifically, the location (latitude and longitude) can be read from the GPS sensor, heading (azimuth) from the magnetometer, and tilt (declination) from the accelerometer. The elevation of the user is queried from the underlying DTM, also stored in the spatial database, based on their lat/long location.

B. 3DQ Server-Side Design

In relation to server-side design of 3DQ, the spatial query processor is divided into 2D and 3D based modules, as shown in Fig. 2.

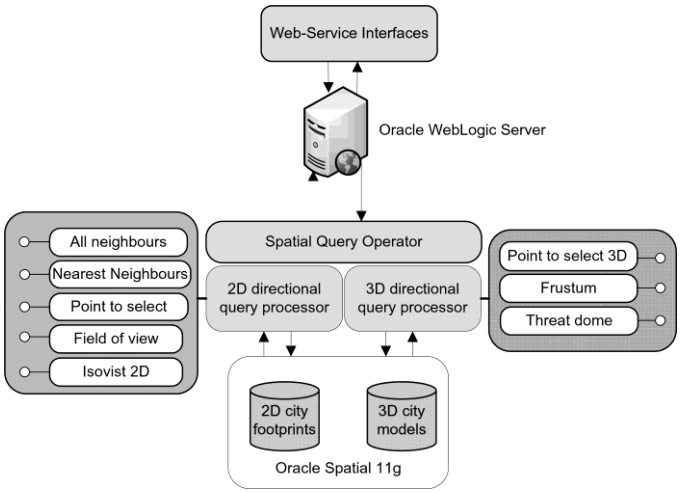


Figure 2. Modules and Interfaces on 3DQ Server-Side

An *Oracle WebLogic* application server is used for deploying the web-services and for providing interfaces to the various mobile devices. Table 1 illustrates the differences in client/server communication complexity using a structured SOAP request, where both the request from a mobile device and the response returned from the server are wrapped in an XML document, and the RESTful formatted query, where a simple URL client request and JSON server response are all that is required for executing the same query.

When compared to XML based SOAP interfaces, the RESTful interfaces are obviously much easier to process on the client-side as each request is simply a URL, while the response in JSON is also more concise and readily parsed by the client. And since working with RESTful decreases the amount of internet data to transfer, this has the added benefit of reducing any potential data transmission costs.

TABLE I. CLIENT-SERVER COMMUNICATION PROTOCOLS

SOAP Interface Request		**RESTful Interface Request**	
XML Sent	***XML Response***	***URL Sent***	***JSON Response***

<SOAP envelope>
...
<Request>
<type>
POIs
<location>
<latitude>53.3387</latitude>
<longitude>-6.2675</longitude>
</location>
<heading>44.04 </heading>
<tilting>20.87</tilt>
</type>
...
</Request>
</SOAP envelope>
	<SOAP envelope >
	...
	<Response>
	<Result>
	<item>
	<name>Jacobs factory</name>
	<coordinate>
	-6.2668, 53.33
	</coordinates>
	</item>
	<item>
	...</item>
	...
	</Result>
	</Response>
	</SOAP envelope>
		http://3DQ.webhop.org/POI/format=JSON&lat=53.33&lng=6.26heading=44.04&tilt=20.87	{"Result":{
			name:"Jacobs factory"
			coordinate:{
			lat:" 53.3321"
			lng:" -6.2668"}
			}}

III. 3DQ SMARTPHONE CASE STUDY

This section presents our prototype iPhone 3GS smartphone with 3DQ installed and performing real-time directional queries in Dublin City. The mobile device comes equipped with an integrated GPS receiver, magnetometer, and accelerometer as standard. Google Maps is used as the background map for displaying query results. The returned result is overlaid on the map in the form of building outlines or POIs to offer the user feedback on their pointing abilities and/or an opportunity to adjust their pointing gestures and re-query if necessary. Some of the more advanced query functions are described below.

A. 3DQ in 2D

This module utilizes a combination of visibility analysis and directional query processing techniques supported by *Oracle Spatial 11g*. The entire information query and retrieval process is performed on the server-side, while the primary parameters required, location and direction, are provided by the mobile device. In addition to typical range and pointing queries already found in some mobile mapping applications today, 3DQ further provides Field-of-View and full Isovist 2D views for performing egocentric visibility based queries. Using a FOV query, only those objects intersecting a user's actual visual field in a particular direction get retrieved (Fig. 3) while a 2D Isovist query will retrieve information about all visible objects in all directions, accurately representing what a user can physically see in 360º from their current location (Fig. 4). The algorithm we use for constructing the query shape is based on [14], which is an open source implementation for efficient and accurate visibility calculations. Fig. 4(a) shows a result set of buildings that intersect with a user's visibility polygon (Isovist) – note the pins get placed by default at a building's centroid as it is stored in the database; and Fig. 4(b) shows a different result set as the user changes location and thus changes the shape of their Isovist. This example illustrates the

process of reducing information overload by invoking hidden query removal (HQR) functionality.

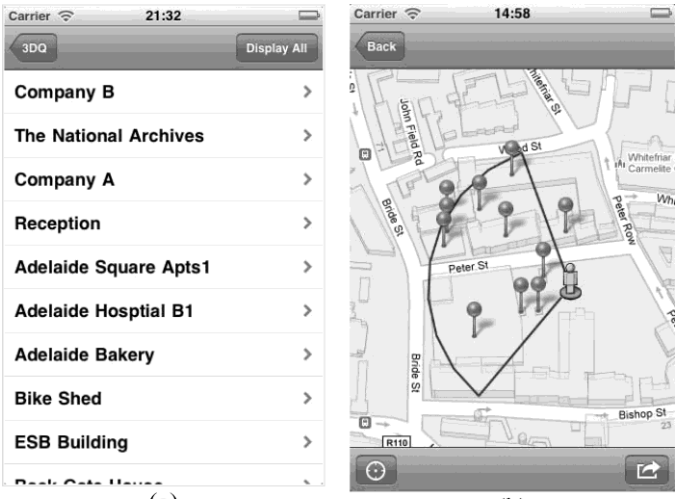


Figure 3. (a) List of all buildings present in user's Field-of-View (b) Selected buildings and FOV query window displayed on the mobile map

Figure 4. (a) Sample Isovist view and its interacting buildings (b) Isovist shape changes dynamically according to changes in user's position

B. 3DQ in 3D

Implementing 3DQ in a 3D context is considerably more complex than in 2D. This is due to the pointing parameters including now the tilt angle of the client device, and importantly the buildings being stored as 3D solids in the database instead of 2D polygons. In contrast to Point-to-Select 2D and depending on the granularity of the dataset, Point-to-Select 3D can tell the 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 or door it is pointing at on that floor (Fig. 5). In 3D, the elevation of the user is also taken into account. If the user is pointing over a lower building to a higher one behind, the 3DQ query processor is capable of recognizing this difference.

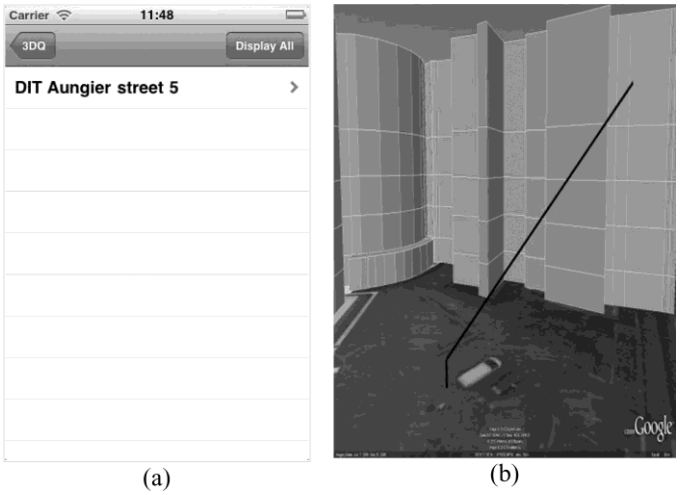


Figure 5. (a) Returned result shows the mobile device is pointing at the 5th floor of a building. (b) Visualization of the pointing ray interacting with building

If a user requires detailed information about individual objects in their field-of-view in a 3D context, a Frustum View query is available to generate the required "flashlight beam" query shape to scan a building and retrieve information on any objects it "illuminates" (Fig. 6).

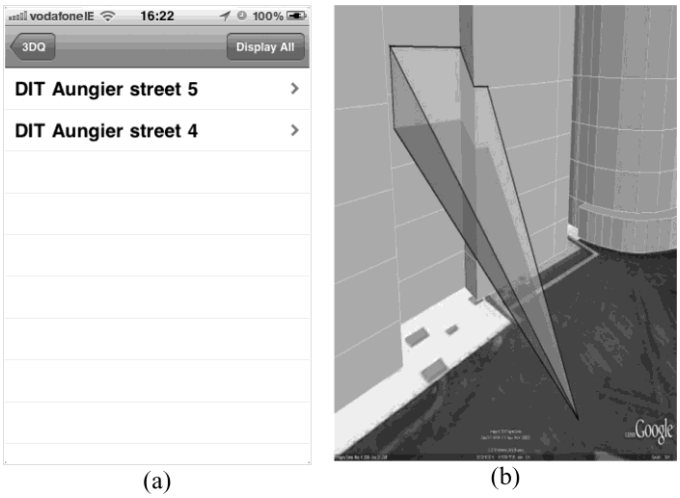
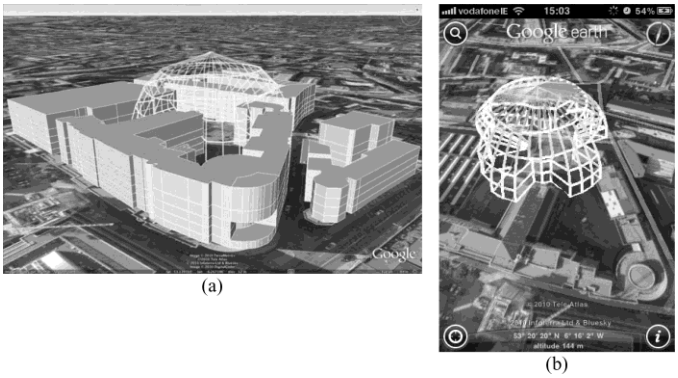


Figure 6. (a) Returned results showing two floors interact with query frustum (b) Visualization of frustum query shape interacting with 3D buildings in the database. Note how HQR redraws the frustum query shape depending on the spatial geometries (building shapes) it interacts with

Finally, a Threat Dome query generates a visibility sphere of 360º in all directions horizontal and vertical, and retrieves information about all database objects/sensor enabled *things* that intersect with the "solid" domed shape (Fig. 7). One example of its use is when a user wishes to retrieve/analyse all visible POIs/environmental sensors (e.g. ATMs, cafés, touristic highlights, CO_2/NO_2/UV readings, noise levels, volcanic ash levels, radiation levels, etc.) at a particular location. This allows for user specific situation awareness type queries such as "show me the cleanest/quietest route through this space". In future work, we intend to include map morphing visualisations to warp the map displayed directly on the device by redrawing the cleanest route as the shortest route, etc.



(a) (b)

Figure 7. (a) An example of a threat dome query shape and real-world 3D building models displayed on a desktop PC (b) The same 3D threat dome displayed in 2D Google Earth using today's limited display functionality on an iPhone.

The described query processes are all implemented by generating the respective query shapes as 3D objects in a 3D indexed spatial database and then merging and segregating outputs from inherent 3D query operators designed to discover any topological relationships (e.g. touch, overlapping, etc.). Currently there are only a limited number of spatial operations available in Oracle Spatial 11g for determining spatial relationships among 3D geometries. These include creating 3D R-Tree indexes on 3D geometries using a minimum-bounding cube. Oracle then considers if these cubes intersect with one another as a method for determining whether the underlying 3D geometries intersect. However, retrieving the actual 3D location where two geometries intersect in 3D is not supported. For example, an important feature of 3DQ is to detect exactly where the intersection between a generated ray (simulating the pointing direction of a smartphone) and a 3D building occurs. In this case, Oracle derives the intersection point using the 2D spatial operator *SDO_INTERSECTION* by first projecting the source query shape (3D ray in this case) and the target (3D building) onto the ground plane and only returning the 2D position of the ray/building intersection. Using this information and combining it with the tilt angle and the 2D distance to the nadir of the actual intersection point, we are left to compute the actual 3D intersection point of this query ourselves.

In an accurately computed threat dome the generated dome will usually have a large number of surfaces relative to the complexity of the surrounding environment. If we want to consider this dome as a single 3D query shape in the form of an Oracle *SDO_GEOMETRY* (in order to make use of the *SDO_INTERSECTION* query operator), we find that its total number of surfaces will typically exceed the number of surfaces allowed in the *SDO_ELEM_INFO_ARRAY* where an arbitrary maximum of 999 coordinates (i.e. 333 3D points) are permitted. In our case, where each element of the surface contains 12 coordinates in its shape (4 vertices), a maximum of only 27 surfaces are allowed in one *SDO_ORDINATE* list – which is far fewer than what is typically required to accurately define a complete threat dome shape. In order to circumvent this limitation, a workaround was devised to first split the complete threat dome into 3 sections and query them individually against the database instead of creating a single 3D solid as a threat dome query shape. The returned query results are then a sum of all object/section intersections after first removing any duplication. Ironically, one beneficial consequence of this extra processing is that it encouraged us to mirror the spatial database across three servers and then send each individual 3D dome section query to a separate database. The result is a much faster (~5sec.) query process which potentially allows for near real-time threat dome visualisations on a mobile device. For this functionality we await the release of the *Google Maps 5* API for Android (summer 2012) allowing for custom 3D mobile map displays.

IV. Conclusions and Future Work

In tomorrow's Web 4.0, trillions of micro-sensors will be installed throughout the world capable of monitoring all types of environmental conditions, events, and processes. As each senor has a unique ID and known coordinates, this Internet of Things will be accessible from anywhere on the Web with each recorded event available for discovery and analysis through intelligent spatial queries. Our mobile three dimensional query (3DQ) prototype is designed to provide dynamic 2D and 3D visibility based directional query services for exploring this myriad of spatial/POI/sensor-web datasets found throughout a built environment on today's location and direction aware smartphones. Moreover, the system employs an open web-service orientated architecture that enables most COTS devices to Connect and Play through their preferred web-browser interfaces regardless of differences in their proprietary operating systems. With 3DQ, a user with a sensor enabled (i.e. GPS, magnetometer and accelerometer) smartphone can interact with the surrounding environment and retrieve information using a combination of pointing gestures and visibility parameters.

The primary requirement for performing accurate directional queries is location, regardless if it's in a 2D or 3D context. Due to significant battery consumption while using integrated GPS, the sensor is not typically powered to a high resolution when compared to GPS receivers for surveying or military use. Therefore improving the accuracy of positioning data (latitude, longitude, and elevation) is a major challenge for further work and may involve trilaterating Wi-Fi or other radio signals (e.g. GSM, Bluetooth, etc.) to improve positioning accuracy [15]. For example, when a user is standing on a second floor balcony, the height (elevation) of the user's current location is needed for accurate HQR results (e.g. a 3D directional query may look over lower objects that from ground level would have obstructed the FOV). Alternatively, by utilising newer touch screen functionality found in some recent smartphones, users could potentially just draw their own query shapes directly on the device at any location they choose – thus not expending any battery power on sensor readings whatsoever [16].

In relation to 3D query shapes, a new approach will investigate using the FOV parameters from the 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 on the same display. Such an approach could be considered a 3D extension to current 2D augmented reality work in this area with the

added advantage of allowing for selecting returned objects/things directly off the same touchscreen display [17,18].

Another useful extension will migrate the 3DQ application to an indoor environment (e.g. inside university buildings, museums, shopping centres, etc.). This will require investigations into accurate indoor positioning that trilaterate a smartphone's inherent ultrasonic capabilities as described in [19].

Arising together with the popularity of today's Web 2.0, a growth in social network type applications that connect people everywhere has been described as increasing exponentially [20]. As "connecting and sharing" become the main activity, every individual (e.g. facebook/twitter user) become themselves a "sensor" by way of forwarding their current events/status to others. Therefore, monitoring a cluster of "citizen as sensors" in the social network could be considered as corresponding to actual micro-sensors in the usual Future Internet context. For instance, in the case of an earthquake, micro-sensors would detect the change in ground vibration and noise levels while the "citizen sensor" would detect changes in their own individual circumstances – whatever they may be. Since human activity is directly involved, a common knowledge (i.e. trend) could be generated and combined with any raw data recorded by the micro-sensors – together making available yet another potentially very important resource for thematic spatial data mining.

Further testing of 3DQ will be carried out on a geometrically detailed NUI Maynooth 3D Campus model [21] constructed from LiDAR data and linked to university specific meta-data (e.g. class schedules, office/lab locations/times, etc.). It is envisioned that these tests will help prepare 3DQ for extension to any Future Internet datasets provided through ongoing Smart City developments in Dublin and elsewhere. Communicating existing and future onboard smartphone sensor data (e.g. noise/light levels) to the server will adhere to OGC Sensor Web Enablement (SWE) standard encodings for measured sensor observations (O&M) and, where appropriate, various web-services (e.g. SOS, SAS, SPS) for providing access to sensor descriptions (encoded in SensorML) and real-time alerting and tasking of sensors or sensor systems as described in [22,23] and in extensive implementations of these standards developed most notably by [24].

REFERENCES

[1] Ball, M. (2011): "How do crowd sourcing, the Internet of Things and Big Data converge on geospatial technology?" in: V1 Magazine, Vol 5, Issue 41, October 2011

[2] P. Frohlich, M. Baldauf, P. Reichl and R.F. Tobler, "Visual Presentation Challenges for Mobile Spatial Applications: Three Case Studies" in: Information Visualisation IV '08 12th International Conference, IEEE Xplore 1550-6037/0, London, July 2008, pp.533-538.

[3] K. Gardiner and J.D. Carswell, "Viewer-based Directional Querying for Mobile Applications" in: 4th International Workshop on Web & Wireless Geographical Information Systems (W2GIS2003), IEEE CS Press, Rome, December 2003

[4] S. Di Martino, F. Ferrucci, G. McArdle, G. Petillo, "Automatic Generation of an Adaptive WebGIS" in: 9th International Symposium on Web & Wireless GIS (W2GIS2009), J.D. Carswell et al. (Eds.), Springer LNCS vol. 5886, Ireland, December 2009, pp.171-186

[5] S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai and M. Matera, "Designing Data-Intensive Web Applications" in: The Morgan Kaufmann Series in Data Management Systems, Morgan-Kaufmann Publishers, ISBN-13: 978-1558608436, San Francisco, December 2002

[6] E. MacAodih, D. Wilson, M. Bertolotto, "A Study of Spatial Interaction Behaviour for Improved Delivery of Web-Based Maps" in: 9th International Symposium on Web & Wireless GIS (W2GIS2009), J.D. Carswell et al. (Eds.), Springer LNCS vol. 5886, Ireland, December 2009, pp.120-134

[7] D.M. Mountain, "Spatial Filters for Mobile Information Retrieval" in: 4th ACM Workshop on Geographical Information Retrieval (GIR'07), Publisher: ACM Press, Lisbon, November 2007, Pages: 61-62

[8] Sig.ma, "Semantic Information Mashup" in: http://sig.ma, [Accessed October, 2011]

[9] D. Evans 2010, "Cisco Futurist Discusses Internet of Things" in: http://blogs.cisco.com/news/comments/-cisco_futurist_discusses_internet_of_things_tech_predictions_more_on_talk2c/, [Accessed March, 2011]

[10] EU Commission 2010, "ICT Research in FP7" in: ICT - INFORMATION AND COMMUNICATION TECHNOLOGIES, pp25, ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/ict-wp-2011-12_en.pdf, [Accessed March, 2011]

[11] M. Ball, 2011, "How Does the Gap Between Sensors, Systems and Analysis Get Filled" in: V1 Magazine, http://www.vector1media.com/dialog/perspectives/19334-how-does-the-gap-between-sensors-systems-and-analysis-get-filled.html, [Accessed April, 2011]

[12] M. Kalin, "Java Web Services: Up and Running" in: O'Reilly Media Publishers, ISBN-13: 978-0596521127, Februrary 2009

[13] R. Simon, P. Frohlich, "A Mobile Application Framework for Geospatial Web" in: 16th International Conference on World Wide Web, Banff, Alberta, May 2007, pp.381-390

[14] K.J. Obermeyer, "The VisiLibity Library" in: http://www.VisiLibity.org, 2008,[Accessed March 1, 2011]

[15] S. Rooney, K. Gardiner, J.D. Carswell, "An Open Source Approach to Wireless Positioning Techniques" in: The 5th International Symposium on Mobile Mapping Technology (MMT'07); Padua, 2007

[16] J. Yin, J.D. Carswell, "Touch2Query Enabled Mobile Devices: A Case Study Using OpenStreeetMap and iPhone" in: 10th International Symposium on Web and Wireless Geographical Information Systems (W2GIS'11), Springer LNCS vol. 6574, pp. 203-218, Kyoto, March 2011

[17] CAMINEO, "The Multimedia Guide for Tourism" in: http://www.layar.com/, [Accessed April 15, 2011]

[18] Layer, http://www.layar.com/, [Accessed April 15, 2011]

[19] V. Filonenko, J.D. Carswell, "Hybrid Indoor Positioning and Directional Querying on a Ubiquitous Mobile Device" in: The 6th International Symposium on LBS and TeleCartography, CGS, University of Nottingham, September 2009

[20] M.F. Goodchild, "Citizens as Sensors: the World of Volunteered Geography" in: GeoJournal (2007) 69:211-221, DOI 10.1007/s1078-007-9111-y

[21] NUIM 3D campus model, in: StratAG, http://www.stratag.ie , 2009, [Accessed May 1, 2011].

[22] M. Botts, G. Percivall, C. Reed, J. Davidson, "OGC Sensor Web Enablement: Overview and High Level Architecture" in: OGC White Paper, Open Geospatial Consortium Inc., OGC 07-165, 28, December 2007

[23] S. Jirka, A. Broering, A. Walkowski, "Sensor Web in Practice" in: GEO Informatics, September 2010

[24] 52North in: http://www.52north.org, [Accessed April 19, 2011]