Modelling and Handling Seams in Wide-Area Sensor Networks

Joseph Newman[12], Gerhard Schall[1] and Dieter Schmalstieg[1]
[1]Institute for Computer Graphics and Vision, Graz University of Technology
[2]Computer Laboratory, University of Cambridge
{newman, schall, schmalstieg}@icg.tu-graz.ac.at

Abstract

The fields of Wearable Computing, Augmented Reality and Ubiquitous Computing are in principle highly convergent, as they all promise a utopian future in which the devices embedded in the environment, our bodies and our clothes will have reached a level integration such that we can intuitively perceive and interact with our environment. However, the reality as practised in research labs and limited commercial deployments has been that budgetary and technical constraints have actually kept these fields separate and distinct. One manifestation of this separation is in the choice of sensors used to build systems in each domain. A truly cross-disciplinary project has to incorporate sensors of much greater heterogeneity than has occurred heretofore. The way in which sensors are deployed results in spatial seams that can act as obstacles to the provision of services across different areas. This paper takes an architectural approach to handling events from different tracking systems and maintaining a consistent spatial model of people and objects. The principal distinguishing feature is the automatic derivation of dataflow network of distributed sensors, dynamically and at run-time, based on requirements expressed by clients.

1. Introduction and Related Work

Table 1 provides a summary of the characteristics of the different sensors used in Virtual Reality (VR), Augmented Reality (AR) and Ubiquitous Computing (Ubicomp) applications, from which we can draw a distinction between *Tracking* (high bandwidth and high precision) and *Location* which is not necessarily high precision but is globally referenced (to some scale). Middleware architectures have reflected this division favouring either a federated data model such as Nexus [2] and QoSDream [3], or overall throughput to a minimal number of clients such as OpenTracker [11] and VRPN [13].

Cross-disciplinary research efforts that have attempted to unify aspects of AR, Ubicomp and Wearable Computing have inevitably used a mix of sensor types, that have resulted in spatial seams between different areas of coverage. In the AR domain Höllerer et al. [5] first introduced mobility to AR with their MARS backpack setup. Newman and Reitmayr introduced “Wide Area Indoor Sentient AR” [8] and “Outdoor Collaborative AR” [12] respectively, thus introducing Ubicomp concepts.

Hybrid indoor and outdoor tracking systems, such as those of Höllerer [4] and Piekarski [10], have generally relied on a very specific blend of sensors, rather than dealing with the issue of how to exploit the arbitrary range of devices that happen to be deployed. Klinker et al. first postulated that distributed tracking concepts [7] were necessary to propel AR beyond the confines of the laboratory into serious industrial settings, leading directly to the development of the DWARF [1] system.

	VR	**AR**	**Ubicomp**
Update rate	15 — 500 Hz		∼1 Hz
Latency	$<$ 30 ms		∼1 s
Cost	Expensive		Cheap
Working volume	Small ($< 5\mathrm{m}^3$)		Large
Granularity	Fine		Coarse
Quantity	Scarce		Abundant
Tracking	✓	✓	✗
Location	✗	✓	✓

Table 1: Sensor characteristics comparison for Augmented Reality, Virtual Reality and Ubicomp

Table 2 shows the different sorts of events that typically take place in a “Ubiquitous AR” environment, as people and objects move around. The bandwidths of these events suggest different strategies as to how a system should respond. For example, tracker pose has an extremely high bandwidth and should, therefore, be communicated using a very fast, highly optimised system from tracking *sources* to application *sinks*. In contrast, the tracker scope, and dataflow reconfiguration events should be relatively infrequent and thus can be handled by a framework that emphasises flexibility over absolute performance. Accurate surveys of environmental features such as the positions of fiducial markers may happen once a year at most, whilst ad-hoc measurements using simple tools such as tape measures may be more frequent if necessary.

A new type of middleware support is required, which we call Ubiquitous Tracking [9] or Ubitrack. This paper describes how an environment and its resources can be dynamically modelled; client queries expressed and registered; and client needs met insofar as is possible. A comprehensive approach to fusion of sensor data is beyond the scope of this paper, as the error statistics of typical sensors are too complex (e.g. non-linear and non-Gaussian) to respond to a naive approach. This paper focusses on an architectural approach to handling numerous diverse and disparate sensors and clients, albeit based on simple heuristics. The principal contribution of our work is that it provides fully automatic derivation of dataflow networks of distributed sensors dynamically and at run-time, based on requirements expressed by multiple clients.

Event	Description	Bandwidth
Tracker Pose	At timestamp *T*, locatable *A* has pose *P* measured by tracker *B*.	High 15 – 50 Hz
Tracker Scope	Locatable *A* is either being tracked by tracker *B*, or is no longer tracked by tracker *B*	Medium 1 – 0.1 Hz
Data Flow Reconfiguration	The flow of data from trackers to clients is reconfigured.	Low 0.2 – 0.01 Hz
Measurement	Measurement of an environmental property, e.g. points *C* and *D*, are separated by distance *d*.	Very low < 0.01 Hz

Table 2: Ubiquitous AR Event Bandwidth

2. Ubitrack Architecture

The Ubitrack architecture shares the three-tier client-server architecture common to many web services and other applications (see Figure 2). The uppermost tier provides an interface for clients to register themselves, and for spatial queries to be expressed. The middle Ubitrack Server (UTS) layer, is responsible for mediating these queries by referring to the bottommost Spatial Relationship Graph Server (SRGS) layer that acts as a spatial database using a graph structure to model objects (e.g. people and locatable objects) as nodes and spatial relationships as edges.

Client Registration A dataflow graph, in practise a dynamically reconfigurable version of OpenTracker [11], is associated with each and every client. Clients-UTS communication is performed by exchanging XML strings: the client requests specific information from the UTS by issuing a query (see below) expressed as XML, and is then asynchronously notified of new configurations by the UTS. The new configurations are delivered in OpenTracker's XML syntax, and then translated into dataflow graphs by the client dynamically.

The objective of building an OpenTracker configuration is to set up a dataflow network that delivers sensor data from both local and remote sensors directly to the client. All physical sensors are directly connected to clients. The dataflow graphs created by OpenTracker are used by multiple clients — both those in possession of or in need of sensor data — to exchange information directly in a peer-to-peer fashion without involving the UTS. The UTS is merely responsible for updating clients concerning the need to alter their dataflow network.

This client-server communication is much lower bandwidth than the bandwidth of the real-time dataflow network between clients (see first and third rows of table 2). These two levels of communication, real-time sensor dataflow between clients and slower paced configuration messages between client and UTS, are essential to deliver robustness and performance simultaneously. Whilst we have used our own in-house Augmented Reality system to build the client applications, the extensible and open design of OpenTracker makes it easy to use the framework with any kind of application, framework or platform.

The UTS refers to a data repository, using the IP address of the host on which a client resides as a key, to determine what specific capabilities the device has, and that can be used by the system in general and the client in particular. For example the client (Tablet PC) shown in Figure 2a has a webcam, an inertial tracker and a 3DOF position tracker (Ubisense). Pure spatial relationships of these devices required for spatial reasoning are maintained by the SRGS. Other properties (such as networking parameters) are maintained in the UTS, which decorates the results from spatial queries issued to the SRGS to yield descriptions of OpenTracker dataflow that are pushed directly to the clients. For example, a Ubisense device physically attached to a client communicates wirelessly with stationary Ubisense infrastructure rather than directly with the client. Therefore the Ubisense data must be relayed from the infrastructure to the Tablet PC or other clients. The SRGS is able to determine how to track the position of the Tablet PC, while the UTS is able to describe a corresponding dataflow network.

Query Registration Clients issue queries about entities or sensors in the environment to the UTS on a one-off basis, or more usually request continuous notification concerning entities of interest in the form of new configurations. These

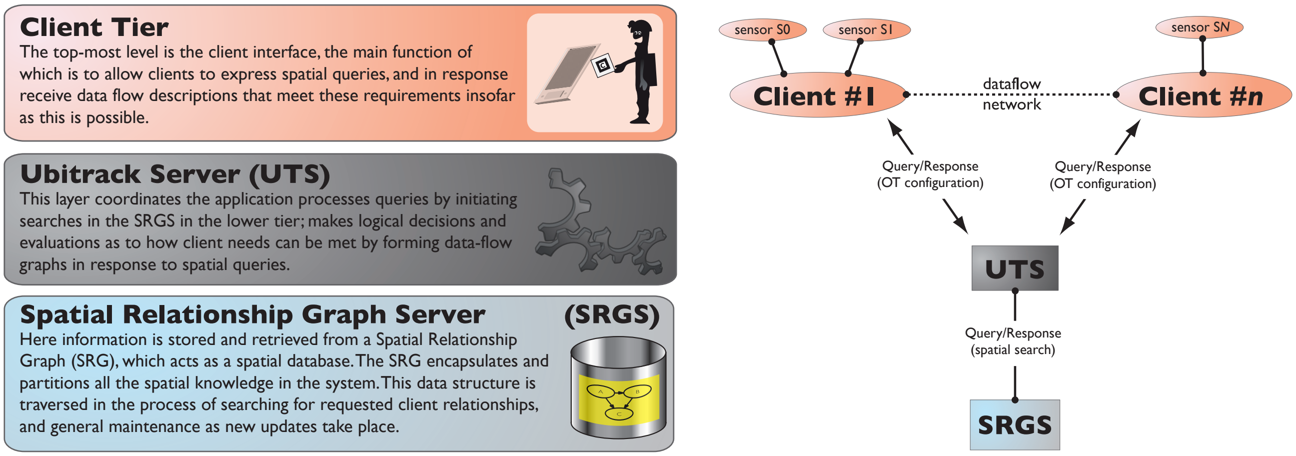


Figure 1: Ubitrack system three tier separation and corresponding deployed architectural structure

queries refer to locales expressed using either spatial primitives (e.g., a room), or as sets, for example the space “second floor but not offices F.3.04 or F.3.05”. These queries are expressed in another XML dialect. The main elements of a query are a description of a spatial relationship between two entities (or groups of entities) in the world.

Locale handover As the mobile user roams the building they will cross between locales, which means that the graph in the SRGS must be updated and the query specified in the previous section re-evaluated. Special OpenTracker nodes sample the position of the mobile devices and moving the device between locales accordingly.

3. Results

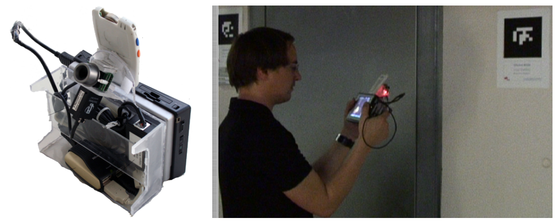
A number of tracking devices were brought to bear on the problem of tracking a mobile user with a Vaio U70 Tablet PC equipped with a webcam, an InertiaCube3 inertial tracker, and a Ubisense device as shown in Figure 2a. The Ubisense ultra-wideband tracking system covers the main laboratory and the adjoining corridors, and is capable of locating many Ubitags to an accuracy of 10-15 cm. Figure 2d shows two figures being tracked, the leftmost one is equipped with the Vaio, and the rightmost one is only tracked by a worn Ubisense device. Accurately surveyed fiducial markers are deployed and be can be seen in Figure 2b, and are visible to the webcam on the Tablet PC, although not necessarily at all times.

This example involves two people, and two applications, which are both versions of the indoor navigation system described in [6]. The mobile user starts in the corridor area, and is tracked using a combination of Ubisense and inertial tracker, or by optical tracking whenever a fiducial marker is visible. The third person desktop view displays the location of both users, at all times throughout the area of interest. However, initially the mobile user only sees his own position in the world-in-miniature view visible in Figure 2c.

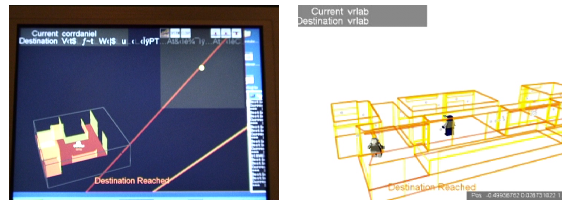
Upon moving into the laboratory area, the change of locale is detected, a change of locale event is relayed to the SRGS, and the queries are re-evaluated by the UTS. Another person in the same locale as the mobile user is found, and a new OpenTracker configuration is pushed to the mobile user reflecting this, and the fact that there is new tracking infrastructure present. While the orientation of the second user is unknown, the Tablet PC user obtains a very good angular response from the inertial tracker. Whenever a fiducial marker is detected by the webcam, the orientation tracker is re-initialized. Note that the novelty of our approach is not in the specific details of this simple sensor fusion, but rather in the fact that this sensor fusion configuration is derived automatically, and updated dynamically.

4. Future Work

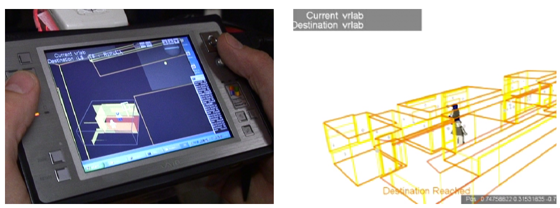
The most important improvement of the Ubitrack approach will be to derive new methods for automatically identifying relevant changes in the world, and deriving suitable dataflow configurations and in particular sensor fusion setups to leverage the available information. More work is also needed to verify the implementation with larger-scale sensor networks and more simultaneous clients. Performance optimisations, such as incremental re-configuration of dataflow graphs without completely halting the dataflow is on the list for future improvements. Moreover, the locale management and membership change detection can be optimised using some form of spatial indexing.



(a) Handheld platform (b) Mobile user equipped with tablet PC



(c) Mobile client with world-in-miniature view (d) Desktop third person view



(e) Mobile client in lab with world-in-miniature view (f) Desktop third person view of both people in lab

Figure 2: Video stills and images as a mobile user moves from a corridor into a lab whilst another person wears a Ubitag

5. Conclusion

Graph structures have been shown to be an effective way of partitioning space, and modelling environments. Queries can be expressed that are continuously reviewed, such that a best effort is made to ensure client needs are always met. The net result is that Ubicomp and AR applications, including legacy applications, can be integrated in ways that were not hitherto possible. This paper has shown that AR has left the confines of the laboratory and reached the corridor, and are moving in a direction that will justify the epithet of "ubiquitous" to AR.

6. Acknowledgements

This work was sponsored by the Austrian Science Fund FWF (contract no. Y193) and the PRESENCCIA Integrated Project funded under the European Sixth Framework Program, Future and Emerging Technologies (FET) (contract no. 27731).

References

[1] M. Bauer, B. Bruegge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor, and M. Wagner. Design of a component-based augmented reality framework. In *Proceedings of the 2nd IEEE and ACM International Symposium on Augmented Reality (ISAR 2001)*, New York, NY, October 2001. IEEE Computer Society.

[2] M. Bauer and K. Rothermel. Towards the observation of spatial events in distributed location-aware systems. In *Towards the Observation of Spatial Events in Distributed Location-Aware Systems*, pages 581—582, Los Alamitos, California, July 2002. IEEE Computer Society.

[3] G. Coulouris. Review report: The qosdream project. Technical report, Laboratory for Communication Engineering, University of Cambridge, 2002.

[4] D. Hallaway, T. Höllerer, and S. Feiner. Bridging the gaps: Hybrid tracking for adaptive mobile augmented reality. *Applied Artificial Intelligence Journal. Special Issue on AI in Mobile Systems*, 18(6):447–500, 2004.

[5] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring mars: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, 23(6):779–785, 1999.

[6] M. Kalkusch, T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann, and D. Schmalstieg. Structured visual markers for indoor pathfinding. In *Proceedings of the First IEEE International Workshop on ARToolKit (ART02)*, Darmstadt, October 2002. IEEE Computer Society.

[7] G. Klinker, T. Reicher, and B. Bruegge. Distributed user tracking concepts for augmented reality applications. In *Proc. of ISAR 2000, Munich*, Munich, Germany, Oct. 2000. IEEE Computer Society.

[8] J. Newman, D. Ingram, and A. Hopper. Augmented reality in a wide area sentient environment. In *Proc. of IEEE and ACM Int. Symp. on Augmented Reality (ISAR 2001)*, pages 77–86, New York, NY, October 2001.

[9] J. Newman, M. Wagner, M. Bauer, A. M. lliams, T. Pintaric, D. Beyer, D. Pustka, F. Strasser, D. Schmalstieg, and G. Klinker. Ubiquitous tracking for augmented reality. In *Proceedings of ISMAR 2004*, November 2004.

[10] W. Piekarski, B. Avery, B. Thomas, and P. Malbezin. Integrated head and hand tracking for indoor and outdoor augmented reality. In *Proceedings of IEEE Virtual Reality Conference*, Chicago, IL, USA, March 2004.

[11] G. Reitmayr and ieter Schmalstieg. Opentracker - a flexible software design for three-dimensional interaction. *Virtual Reality*, 9(1):79–92, December 2005.

[12] G. Reitmayr and D. Schmalstieg. Collaborative augmented reality for outdoor navigation and information browsing. In *Proc. Symposium Location Based Services and TeleCartography 2004*, Vienna, Austria, 2004.

[13] R. M. Taylor, ll, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. VRPN: a device-independent, network-transparent VR peripheral system. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 55–61. ACM Press, 2001.