Fundamentals of Ubiquitous Tracking

Martin Wagner, Asa MacWilliams, Martin Bauer, Gudrun Klinker,*
Joseph Newman, Thomas Pintaric, Dieter Schmalstieg†

Abstract

Ubiquitous Computing (Ubicomp) environments require detailed, coherent and up-to-date spatial models of the world. However, current tracking technologies are limited in their range and operating environments. To extend the scope of Ubicomp applications, it will be necessary to combine heterogeneous tracking and sensor technologies dynamically, aggregating their data and balancing their trade-offs. In this paper, we propose a formal framework, called *Ubiquitous Tracking*, which uses a graph-based model of spatial relationships to build dynamically extendible networks of trackers with high-precision, low-latency requirements. The framework is powerful, allowing us to model existing complex tracking setups; extensible, accommodating new trackers, filtering schemes and optimisation criteria; and efficient, allowing an effective implementation within existing systems.

1. Introduction and Related Work

Any attempt to implement the "calm" [17] technology envisaged by Weiser in his Ubiquitous Computing (Ubicomp) [15, 16] paradigm demands a level of awareness (that does not necessarily imply intelligence) on the part of the system such that appropriate feedback is provided at the periphery of user's consciousness. Sentient Computing [1, 6, 7] is an approach to Ubicomp that maintains a model of spatial relationships, which appears to reproduce the perceptions a user has of the world. Networks of sensors are used to make measurements and estimates of environmental state keeping the model up-to-date.

Augmented Reality (AR) provides a natural mechanism for interacting with a Ubicomp environment. The augmentation of the user's senses can provide an intuitive interface to an unobtrusive and yet enhanced view of the world. Our motivation is derived from the high-precision, low latency modelling required to maintain the illusion of immersion in coexistent real and virtual worlds.

Classic AR applications, such as the Boeing wire assembly feasibility study [4], have been constrained by current tracking technologies to carefully arranged spaces of a few square metres. Systems that have aimed at true mobility, and hence ubiquity, such as the Touring machine [5] and Sentient AR [11] have typically relied on a ubiquitous tracking infrastructure, such as GPS. Wide-area trackers generally provide only modest levels of accuracy at a low update rate, and cannot be used for tasks requiring greater precision. Furthermore, these systems are generally homogeneously deployed throughout the area of interest resulting in much tedious off-line calibration and registration.

*Institut für Informatik, Technische Universität München, {**wagnerm, macwilli, bauerma, klinker**} **@ in.tum.de**
†Vienna University of Technology, {**newman, pintaric, schmalstieg**} **@ ims.tuwien.ac.at**

In real large-scale environments it is probable that the quality of sensing and tracking will vary significantly. There will be a few, small high precision spots where expensive trackers such as those used to track medical instruments in an operating theatre would support a particular task, such as surgery. Conversely, in other areas, like hallways, cheap cell-based tracking may be sufficient. The ad-hoc combination of diverse tracking resources allows an optimal trade-off between precision, hardware cost and complexity to be struck, such that the system can be both effective and affordable. We propose a new approach, *Ubiquitous Tracking*, formalising a mathematical framework in which dynamic qualities of spatial relationships between objects represented in the model can be expressed in a graph. A formal model will provide a common terminology for discussing this subject as well as an integrated approach to accommodating new sensors and performing auto-calibration of complex sensor networks. A full description is beyond the scope of this paper, however interested readers can obtain finer detail in our other work [12].

2. Spatial Relationship Graphs

The goal of the theoretical framework discussed in this paper is to provide a query mechanism that returns an optimal estimate of the geometric relationships between arbitrary objects, according to user-defined criteria. For this purpose, we use a graph-based model of spatial relationships. We first discuss properties of real-world relationships, then consider properties of measurements made in the real world and finally present a general concept of how knowledge can be inferred from these measurements, resulting in the construction and maintenance of a model of the real world.

Fundamentals We visualise the spatial relationships between objects in a graph structure [3] in which nodes represent objects and edges represent spatial relationships between the objects.

A tracking device is simply a sensor that makes measurements of the spatial properties of objects. In order to interpret data appropriately other attributes such as a timestamp and the uncertainty associated with a measurement must also be considered. We distinguish between three different sorts of graph: an idealised view of the world from the point of view of an omniscient observer, directly measured relationships corrupted by noise, and ultimately a graph of inferred relationships derived from measured relationships.

Real Relationships In the real world, each pair of objects has, at every point in time, a geometric relationship that can be expressed using the standard computer graphics notation of a 4×4 homogeneous matrix representing arbitrary transformations between coordinate systems. We define a binary relation Ω on our object space $N = \{A, B, C, \ldots\}$. We then map every element (X, Y) of Ω onto a function w_{XY} describing the spatial relationship between the objects X and Y over time. This attribution scheme is called $\mathbf{W}$.

$$\mathbf{W} : (\Omega = N \times N) \rightarrow w, \text{ where } w : D_t \rightarrow \mathbb{R}^{4\times4} \tag{1}$$

D_t is the source time domain, mapped by w onto the target spatial relationship domain. This definition matches the output of common tracking devices, yielding spatial relationships for different points in time. An omniscient observer would be able to perceive geometrical relationships between all objects and the Ubiquitous Tracking problem would be solved.

Measured and Inferred Relationships Unfortunately, we can only make estimates of geometric relationships between real objects by taking *measurements*. Each measurement is made at a discrete point in time, yielding a geometric relationship that is equivalent to the real relationship, but corrupted by noise. The quality of measurements is described using a set of *attributes*, $\mathcal{A}$, which includes properties such as latency, confidence values, or a standard deviation in meters. By analogy to the relation Ω, we now define a relation Φ and an attribution $\mathbf{P}$ describing the measurements:

$$\mathbf{P} : (\Phi \subseteq N \times N) \rightarrow p, \text{ where } p : D_t \rightarrow \mathbb{R}^{4\times 4} \times \mathcal{A} \tag{2}$$

Figure 1 illustrates an optical shared tracking setup [8] and associated graph, $G(\Phi)$, providing a visual representation of the relation Φ. An edge between two nodes exists only if measurements have been made. In the shared tracking example, we assume that the geometric relationship between camera A and fiducial marker B as well as between camera D and the markers B and C has been measured yielding functions p_{AB}, p_{DB} and p_{DC} with attribute sets $\mathcal{A}_{AB}$, $\mathcal{A}_{DB}$ and $\mathcal{A}_{DC}$ describing the quality of the measurements.

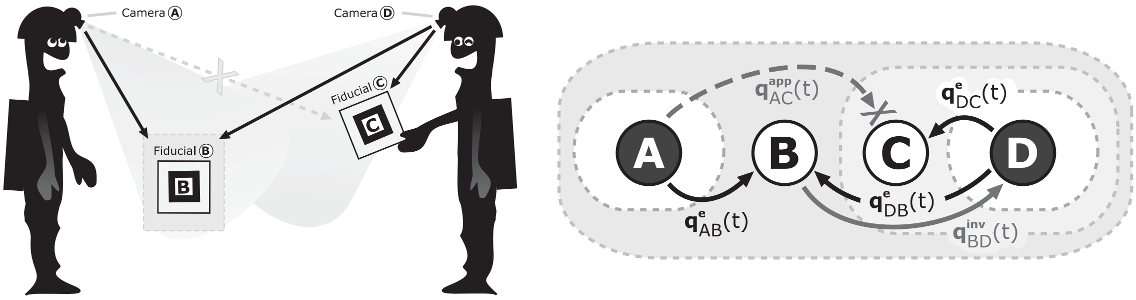

Figure 1. Example setup: Both cameras A and D detect fiducial marker B, but only camera D detects marker C. However, the application is interested in marker C's geometric relation to camera A. On the right, the graph containing all relevant inferred relations is shown. Using this inferred knowledge, we can compute the desired relation q^{app}_{AC}.

The goal of the example discussed here is to obtain at every point in time an estimate of the spatial relationship H_{AC} between camera A and marker C. For this purpose, we introduce a binary relation Ψ and an attribution $\mathbf{Q}$ that initially consists of all measurements stored in relation Φ.

$$\mathbf{Q} : (\Psi \subseteq N \times N) \rightarrow q, \text{ where } q : D_t \rightarrow \mathbb{R}^{4\times 4} \times \mathcal{A} \tag{3}$$

We can now *infer* knowledge from the measurements. Every inference results in a new element of Ψ and consequently a new edge in the graph $G(\Psi)$. From figure 1 we see that AC can be calculated using the relations AB, DB and DC. Note that the source time domain D_t of p is discrete, consisting of a finite discrete set of points in time at which the measurements were made. In consequence, two measurements can only be combined, if they were made at exactly the same time. This is very unlikely, thus we infer new relations AB', DB' and DC' over an extended continuous time domain, yielding new functions q^e_{AB}, q^e_{DB} and q^e_{DC}. The attributes describing the quality of these inferred relations are adjusted to reflect their poorer quality as opposed to those of the discrete measurements. The next step we have to take is to infer the relation BD' from DB'. This can be done by inverting the homogeneous matrix $H_{DB'}$ and calculating an appropriate attribute set $\mathcal{A}_{BD'}$. Note that this attribute set may differ significantly from $\mathcal{A}_{DB'}$. We now infer our target geometric relationship H_{AC} by multiplying together H_{AB}, H_{BD} and H_{DC} and calculating the attribute set $\mathcal{A}_{AC}$.

Attributes and Error Functions Besides describing the actual quality of inferred geometric relationships for applications like accuracy based rendering [9] the main purpose of relationship attributes lies in resolving multiple solutions to a query. We define an error function mapping attributes along a path in the graph describing a possible solution onto a real non-negative value.

$$\begin{aligned} e \;:\; & \mathcal{A}^* \rightarrow \mathbb{R} \\ & \mathcal{A}_i^* \mapsto e(\mathcal{A}_i^*) \end{aligned} \tag{4}$$

This error function provides a metric of the relevance of certain attributes for a given application, and the framework returns the solution of a path along which there is a minimal associated error value. In the above example, we might introduce a third marker E that is visible to both cameras. If we assume that the optical trackers attached to the cameras yield a confidence value for their measurements, we may be interested in obtaining the solution AC with maximum confidence. To achieve this, we define the error function to be the product of inverse confidence values along all edges in a path. The framework then evaluates the error function along the paths $ABDC$ and $AEDC$ and returns the path with minimum error value and thus maximum confidence.

The concepts of attributes and error functions can also be applied to the integration of arbitrary filter schemes for combining the data from multiple sensors. The tradeoffs of an individual filter scheme will be reflected in a new set of attributes (e.g. increased latency and improved accuracy) that lead to potentially different paths chosen by the framework.

3. Future Work

The authors have been working on two frameworks for Augmented Reality systems, Studierstube [14] and DWARF [2]. The formal concept of spatial relationship graphs discussed in this paper maps naturally on the component based aproach of these frameworks. We plan to pursue two separate implementation tracks in parallel, while maintaining interoperability between DWARF and Studierstube. For each implementation, a distinction between the abstract model of the spatial relationships between objects and the actual flow of positional information data between components of the run-time system has to be made. An implementation in Studierstube would generate the data flow graphs in the form of OpenTracker [13] networks, which are then evaluated within the Studierstube run-time environment. Conversely, an implementation in DWARF would use DWARF services as the components in the data flow graph, and combine them dynamically, based on their attributes, using the DWARF middleware [10].

To prove the applicability of the framework in a large building-wide Ubiquitous Tracking setup, we will have to design and implement a simulation environment that allows us to generate synthetic data of virtual trackers observing objects roaming through the environment. The data can serve as input to a reference implementation of the framework. We will then conduct run-time measurements to evaluate the efficiency of the framework, and accuracy analysis to refine the design of attributes, error functions and the object state.

Once the framework has been used to implement real scenarios, we hope to gather information and expertise in order to propose a suitable set of standard attributes. These can then be used to describe new tracking technologies, filtering schemes and enable the development of new application domains such as autocalibration of complex sensor networks.

4. Conclusion

This paper presents a theoretical framework to describe tracking setups in a standard way. The framework consists of relations modelled as graphs, which allow application developers to satisfy tracking demands by defining *attributes* and *error functions*. Attributes characterise sensor measurements, and are operated on by error functions that are designed to discriminate between different solutions to a spatial query on a per-application basis.

The framework is a first step towards a systematic implementation of Ubiquitous Tracking concepts and hopefully serves as a basis for partial standardisation of complex tracker setups. Modelling complex setups in a unified mathematical framework brings up new issues commonly overlooked in real-world applications, such as the role of time in processing sensor information. We hope to have initiated a more formal approach to making Ubiquitous Computing environments realisable.

Acknowledgements

The authors would like to thank Gerhard Reitmayr for his valuable contributions, and inspiring work on OpenTracker. This work was partly sponsored by the *High-Tech Offensive Bayern* of the Bayerische Staatskanzlei and by the Austrian Science Foundation *FWF* under contracts no. P14470 and Y193, and Vienna University of Technology by *Forschungsinfrastrukturvorhaben TUWP16/2002*.

References

[1] M. ADDLESEE, R. CURWEN, S. HODGES, J. NEWMAN, P. STEGGLES, A. WARD, and A. HOPPER, *Implementing a Sentient Computing System*, IEEE Computer, 34(8), 2001, pp. 50–56.

[2] 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.

[3] B. BRUMITT, J. KRUMM, B. MEYERS, and S. SHAFER, *Ubiquitous Computing and the Role of Geometry*, IEEE Personal Communications, October 2000, pp. 41–43.

[4] D. CURTIS, D. MIZELL, P. GRUENBAUM, and A. JANIN, *Several Devils in the Details: Making an AR App Work in the Airplane Factory*, in First IEEE Workshop on Augmented Reality, November 1998.

[5] S. FEINER, B. MACINTYRE, T. HÖLLERER, and T. WEBSTER, *A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment*, in Proc. of ISWC 1997, October 1997.

[6] A. HARTER, A. HOPPER, P. STEGGLES, A. WARD, and P. WEBSTER, *The Anatomy of a Context-Aware Application*, in Mobile Computing and Networking, 1999, pp. 59–68.

[7] J. HIGHTOWER, B. BRUMITT, and G. BORRIELLO, *The Location Stack: A Layered Model for Location in Ubiquitous Computing*, in Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), Callicoon, NY, June 2002, pp. 22–28.

[8] F. LEDERMANN, G. REITMAYR, and D. SCHMALSTIEG, *Dynamically Shared Optical Tracking*, in 1st Int'l Augmented Reality Toolkit Workshop, 2002.

[9] B. MACINTYRE, E. M. COELHO, and S. JULIER, *Estimating and Adapting to Registration Errors in Augmented Reality Systems*, in IEEE Virtual Reality Conference 2002 (VR 2002), Orlando, Florida, March 2002.

[10] A. MacWilliams, T. Reicher, and B. Brügge, *Decentralized Coordination of Distributed Interdependent Services*, in IEEE Distributed Systems Online – Middleware Work in Progress Papers, Rio de Janeiro, Brazil, June 2003.

[11] 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), New York, NY, October 2001, pp. 77–86.

[12] J. Newman, M. Wagner, T. Pintaric, A. MacWilliams, M. Bauer, G. Klinker, and D. Schmalstieg, *Fundamentals of Ubiquitous Tracking for Augmented Reality*, Tech. Rep. TR-188-2-2003-34, Vienna University of Technology, 2003. `http://www.ims.tuwien.ac.at/publication_detail.php?ims_id=TR-188-2-2003-34`.

[13] G. Reitmayr and D. Schmalstieg, *OpenTracker – An Open Software Architecture for Reconfigurable Tracking based on XML*, in Proceedings of the ACM Symposium on Virtual Reality Software & Technology (VRST), Banff, Alberta, Canada, 2001.

[14] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavari, L. M. Encarnao, M. Gervautz, and W. Purgathofer, *The Studierstube Augmented Reality Project*, PRESENCE - Teleoperators and Virtual Environments, 11(1), pp. 32–45.

[15] M. Weiser, *Hot Topics: Ubiquitous Computing*, IEEE Computer, October 1993.

[16] M. Weiser, *Some Computer Science Problems in Ubiquitous Computing*, Communications of the ACM, July 1993.

[17] M. Weiser and J. S. Brown, *Beyond Calculation: The Next Fifty Years of Computing*, Copernicus, 1998.