Location of Mobile Devices Using Networked Surfaces

Frank Hoffmann and James Scott

Laboratory for Communications Engineering
Cambridge University Engineering Department
fh215@eng.cam.ac.uk, jamesscott@acm.org

Abstract. Networked Surfaces are a novel technology, using contact with physical surfaces such as desks to provide network connectivity for mobile devices. In addition, Networked Surfaces can accurately estimate the positions and orientations of connected devices, combining two key technologies for ubiquitous computing. This paper discusses the algorithm implemented to estimate device locations on the Networked Surface prototype. It then evaluates the location accuracy obtained, using simulations, measurements, and visualisation. Methods of improving the location accuracy are also investigated. Finally, the paper discusses how the location information provided by Networked Surfaces can be incorporated into context-aware computing applications.

1 Introduction

Context-aware computing[1] applications aim to increase the benefits of using computing devices by enabling them to automatically respond to changes in their environment. The location of devices plays an important role in such applications. This paper discusses how *Networked Surfaces* can be used to locate devices.

Networked Surfaces are a novel technology, designed to provide mobile devices with services such as data and power connectivity, while also facilitating device mobility and operating transparently to users. This is achieved by augmenting physical surfaces, such as desks or conference tables, with conductive pads. Devices with a Networked Surface interface, known as *objects*, are also augmented with pads. When an object is placed on a Surface[2], electrical connections are formed automatically using the pads, connecting the object to the services provided by that Surface. The technology exploits the fact that many devices are placed on surfaces like desks while they are in use.

While providing networking and other services, Networked Surfaces can also be used to locate devices placed on them. This is useful as location information is a key technology for a wide range of context-aware computing applications, such as "desktop teleporting" and intelligent environments.

This paper is structured as follows. First, an overview of Networked Surfaces is given, followed by a detailed explanation of the method used to locate objects. This method is then evaluated using simulations and measurements. Ways of improving the

[1] Also known as "Sentient Computing" [11].

[2] "Surface" is used as an abbreviation for "Networked Surface"

location accuracy are also presented, followed by a comparison of the accuracy of this system to other location systems. The paper ends by discussing context-aware applications which can make use of Networked Surface-based location information, and the methods by which such information can be transferred.

2 Networked Surfaces

Although a detailed discussion of the Networked Surface implementation is not the subject of this paper, an overview of the system will be presented here. More details can be found in [21][3].

2.1 System Architecture

Figure 1 shows a diagram of the system architecture. The data networks and power cabling are distributed within the Surface in the form of *function buses*. The function buses are connected to the *surface manager*, which acts as the gateway to other networks.

In order to provide objects with access to the function buses, the Surface is covered with conductive pads (also called "strips"), which can be electronically switched to form a connection or *link* to one of the functions. For scalability reasons, these strips are not centrally controlled, but are instead grouped into *tiles*, with each tile controlled by an autonomous *tile controller*, which is responsible for detecting objects placed on its tile. All tile controllers are centrally managed by the surface manager using the *tile control bus*.

The bases of objects are equipped with conductive pads, which are controlled by an *object controller*. When an object is placed on the surface, the tile controllers and object controller execute a *handshaking protocol*, negotiating the object's connection to the services it requires. After a connection is established, the *object manager* acts as a network interface card (NIC) for the object.

Networked Surfaces are capable of providing different types of data networks, tailored for different classes of devices. In the prototype, a low bit-rate network is implemented for peripherals or sensors placed in the user's environment. Also, a number of high bit-rate networks are provided for objects like laptops. Preliminary measurements are described in [10]. Improvements made since then show end-to-end data rates of up to 5Mbit/s across the high speed data networks.

2.2 Topology

The layout of surface strips and object pads is called the *topology*. The topology must guarantee that when an object is placed on the Surface, an appropriate number of *links* can be established, depending on the services it requires. The connection must be made reliably, regardless of the position and orientation of the object.

A diagram of the topology used in the Networked Surface prototype is shown in Fig. 2. In this topology, the surface pads have the shape of rectangular strips with small

[3] See `http://www-lce.eng.cam.ac.uk/`

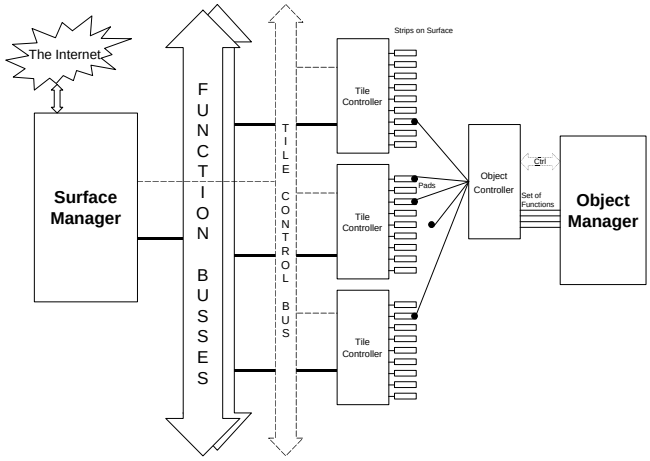

Fig. 1. Networked Surface system architecture

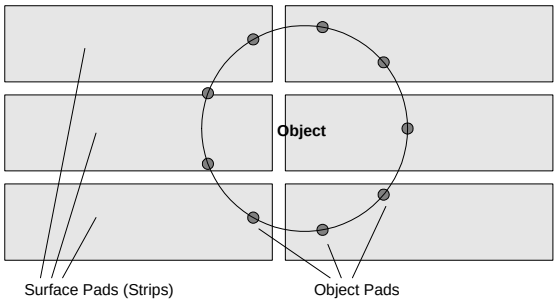

Fig. 2. Example of a Networked Surfaces topology

gaps between them, and the object pads are circular, with their diameter smaller than the gaps between the strips (this prevents the short-circuiting of two adjacent strips). The object pads are arranged in a circle.

Adjusting the number of the object pads and the diameter of the circle allows the creation of various object “footprints,” each of which guarantees a different number of links to be established. This means that the same topology scheme can be used for different types of objects. Figure 3 shows the dimensions of the topology used in the prototype.

As will be seen later, the particular shape of the surface strips and the number of object pads influence the accuracy of the location estimates provided by the Surface.

Surface Topology Details	Object Topology Details		
	Links Required	Object Pads Required	Footprint Diameter (mm)
Tile Size : 250mm x 250mm 24 strips, organized in 2 columns of 12 strips Strip size : 122mm x 17.58mm Gap size : 3mm	2	5	26.4
	3	9	46.2
	4	12	67.8
	5	16	88.2
	6	19	108

Fig. 3. Networked Surface topology details

2.3 Object Connection

When an object is placed on a Networked Surface, the tile and object controllers execute a *handshaking protocol*, during which the object instructs the tiles to connect it to the desired functions. The protocol was designed to connect objects as quickly as possible; early experiments described in [10] show that in 98% of the cases connection is achieved in under one second. Recent work has improved this time to half a second.

For the purpose of locating devices it is important to note that, after the connection of an object, the surface manager has knowledge of which object pads were used for creating the connection, and of the surface strips associated with them. This information is known as a *pad mapping* and can be used to generate a location estimate for that object, as discussed below.

3 Location of Devices on Networked Surfaces

The term "location" is used in this paper to indicate the position and the orientation of the *object origin*, which is defined as the centre of the circle formed by the object's pads. The position of the object origin can be expressed by giving its coordinates in the x and y dimensions. The orientation can be expressed by an angle θ between an arbitrary "up" vector on the Surface and another vector describing the "direction" of the object.

3.1 Location Process

Figure 4 depicts the location process. The location algorithm takes as input a *pad mapping*, expressing which of the object pads is in contact with which of the surface strips, and is a result of the handshaking protocol. In addition it uses knowledge about the topology. The algorithm returns worst case ranges for x, y, and θ as well as a "best" estimate of the object's location within those ranges.

3.2 Location Algorithm Design

One approach to estimating an object's location would be an exhaustive search across x, y, and θ. An exhaustive search would take O(g^3m) time, where g is the granularity

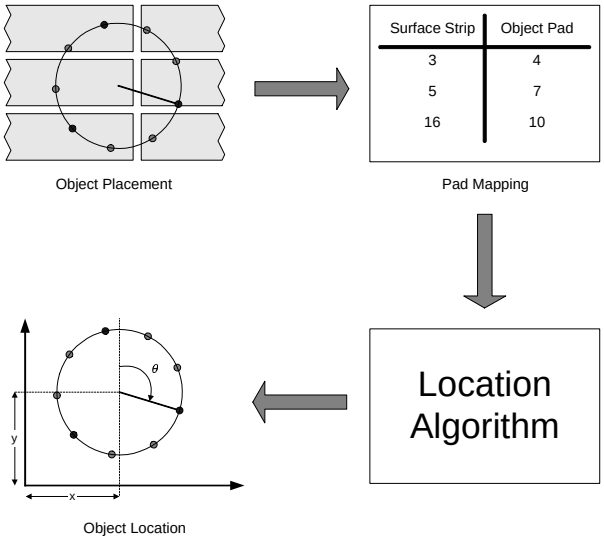

Fig. 4. Object location process

of search used and m is the number of pad mappings provided. In order to be of use in a context-aware system, the location of objects needs to be found as quickly as possible, hence this approach may prove too slow to be useful.

The chosen algorithm avoids the complexity of an exhaustive search by only searching over one dimension, the angle θ. Instead of searching over the remaining two dimensions, the algorithm employs a geometric approach to construct an area in which the object's origin must lie, which is described below. This algorithm allows location to be computed in O(gm) time.

The algorithm iterates over possible orientations of the object, testing whether each orientation is consistent with the known pad mapping. Figure 5 shows the testing process for one orientation. The object is shown as a circle with a line indicating the tested orientation and only the pads used for the connection are shown.

Figure 5(a) shows how the information about one strip (which is touched by one pad p) is used to define a rectangular area that the object origin must be in. A combination of the number of the pad p, the orientation under test and the radius of the object footprint define the vector v describing the position of this rectangle in relation to the strip.

The process is repeated for all links, resulting in the transparent rectangles shown in Fig. 5(b). The orientation under test can only be valid if *all* rectangles intersect. The intersection of the rectangles, shown in Fig. 5(c), is the area in which the object origin must lie for this orientation to be valid.

After iterating over all orientations, a number of valid orientations are found, each with an associated rectangle for which it is valid, as shown in Fig. 5(d). Each point in each rectangle represents a possible location for the given pad mapping. Assuming that all such locations are equally probable, a "best guess" location (x_b, y_b, θ_b) can be found

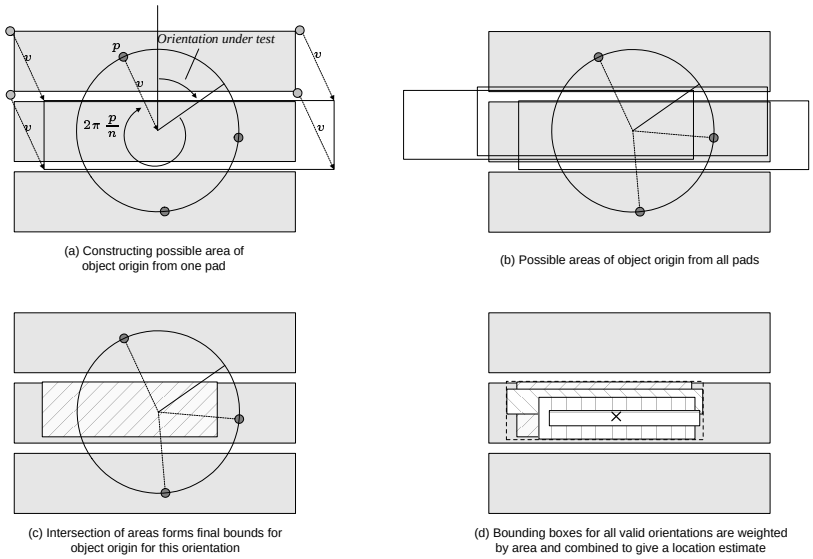

Fig. 5. Testing process for a possible object orientation

using weighted means, as follows:

$$x_b = \frac{\sum_i x_{c_i} A_i}{\sum_i A_i}, \qquad y_b = \frac{\sum_i y_{c_i} A_i}{\sum_i A_i}, \qquad \theta_b = \frac{\sum_i \theta_i A_i}{\sum_i A_i},$$

where (x_{c_i}, y_{c_i}) represents the centre of rectangle i, and A_i represents its area. The (x_b, y_b) position is represented by a cross in Fig. 5(d).

The use of weighted means minimises the average-case error (given the assumption above). If it were instead desirable to minimise the worst-case error, then the best location estimate to use would be the middle of the valid ranges of x, y and θ. The x and y ranges are illustrated by the dotted rectangle in Fig. 5(d). The ability of the location algorithm to define such ranges means that the maximum location error (for a particular Networked Surface topology) is bounded, thereby allowing guarantees of location accuracy to be provided.

4 Evaluation of the Location Algorithm

In order to evaluate the location accuracy in the Networked Surface prototype, both simulations and measurements were carried out. While it is possible to measure the location of a real object placed at many locations on the Networked Surface, this is very time consuming and incurs a considerable measurement error. Because of this, many of the tests were carried out using a simulation. Where possible, the results of the simulation were then verified with a small set of measurements.

Finally, a graphical front end to the location algorithm was implemented, to offer visual feedback of the accuracy of the location algorithm. Results for all three methods are presented and discussed below.

4.1 Simulation Harness

The simulator replaced the process of physically placing a real object onto a Surface by first choosing a random location for an object and then calculating which of the object pads touch which of the surface strips.

The simulator then mimicked the handshaking protocol, in order to obtain a pad mapping. Because of the way the topology is designed, it is possible that more than one object pad touches a single surface strip, and also that the object's pads combined touch more than the required number of surface strips. From all pads that touch one strip, the one with the lowest number was selected; this emulates what the object does during handshaking. If the placement of the object resulted in it touching more than the required amount of strips, the strips used were chosen at random.

The resulting pad mapping, which is identical to the information that would be available after the detection of an object on the prototype, was then passed to the location algorithm, which calculates an estimation of the object's location. Finally this estimate was compared to the original location, and errors were calculated for the variables x, y, the magnitude of the vector (x, y), as well as for the angle θ.

4.2 Simulation

For this test, the simulator determined the location algorithm's accuracy in the dimensions of x, y, the magnitude of the vector (x, y), and θ for one million object locations. Object sizes from two to six links were tested. Figure 6 shows the mean errors recorded and Fig. 7 shows the respective maximum errors.

The figures show that the mean accuracy available depends highly on the number of links used by an object (and hence the number of pad mappings provided). This is especially true of the angular error, whose mean varies from over 35° to under 5° for two to six link objects respectively.

As the shape of the surface strips might suggest, the x error forms the majority of the mean positional error, being about five times the y error; this is also true for the maximum positional errors. However, the maximum errors do not decrease as fast as the mean errors as the number of available pad mappings grows. Indeed, the maximum y error is actually at its lowest in the two-link case. This may be because the location estimate returned is designed to minimise the mean error case, but not the maximum error.

Another result of the simulations was the fact that for all one million trials, the values for x, y, and θ of the estimated location were within the ranges returned for these three variables. This demonstrates the reliability of the algorithm, which always returns an accurate location whenever a connection occurs. It also demonstrates that the algorithm has a bounded maximum error, since the location is never outside the ranges returned.

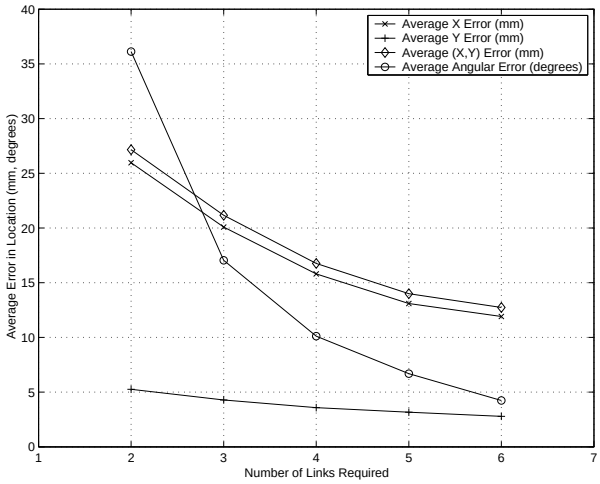

Fig. 6. Mean error in location simulations

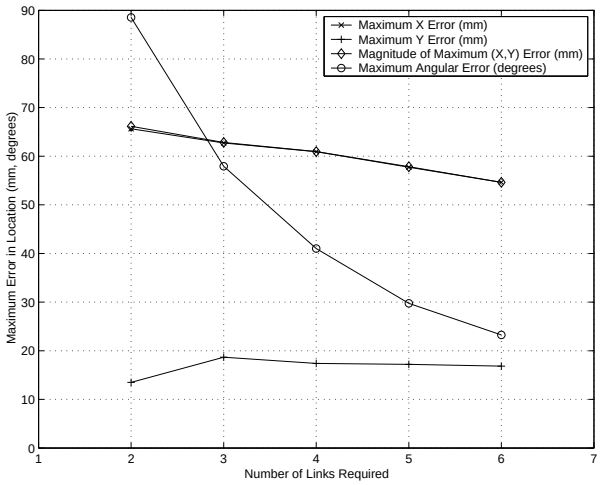

Fig. 7. Maximum error in location simulations

The overall conclusions for location accuracy are that, for a typical four-link object, the Surface prototype will locate the object with a mean error of 1.6cm, and at worst an error of 6.1cm. For orientation, the average error is 10°, and the worst case error is 41°.

4.3 Manual Measurement

For the case of the four-link object, the results from the simulation were verified with a set of measurements carried out on the prototype. An object was placed at fifty random locations on the prototype surface, and its position was measured with a tape measure. The measured location was compared to the one calculated by the algorithm.

Table 1 shows the mean errors found using the manual location measurement compared to the results of the simulation. For all variables, the experimental results were slightly more accurate than the simulated results, but the difference between the respective values is below the estimated accuracy of the manual measurements, which was 5mm. This comparison shows that the experimental results agree with the simulation results. Thus, the simulation is shown to be realistic.

Table 1. Comparison of simulated and experimental results

Variable	Mean Simulated Error	Mean Experimental Error	Difference
x	15mm	13mm	2mm
y	3.6mm	3.0mm	0.6mm
(x, y) vector	16mm	14mm	2mm
θ	7.8°	6.3°	1.5°

4.4 Visual Feedback

In addition to the numerical output of the location algorithm, a graphical front end was developed to offer qualitative feedback of the location results. Figure 8 gives an example of this front end. In this diagram, the circles show the estimated object locations, and the rounded rectangles show the bounds for each object (i.e. one can imagine the circles being moved around, so long as their edges do not cross the rectangles). The three lines within the circles show estimated values for θ, together with the minima and maxima.

By displaying ranges for the position and orientation together with the best estimate, the visualisation tool illustrates how the accuracy differs depending on where an object is placed. High error ranges are generally noticed when the object is situated on one column of surface pads, as can be seen with the object at the top of the diagram. This is due to the much bigger range of possible movement in the x dimension in this situation.

When two columns are spanned, as shown by the object in the bottom left corner, the lowest error ranges are seen. This is because stricter x limits are imposed by the need to be in contact with strips in both columns. Finally, it should be noted that, due to the use of weighted means for the location estimate, the circle is not necessarily centred within the rectangle, and the middle line does not necessarily bisect the other two. This is illustrated by the object in the lower right corner.

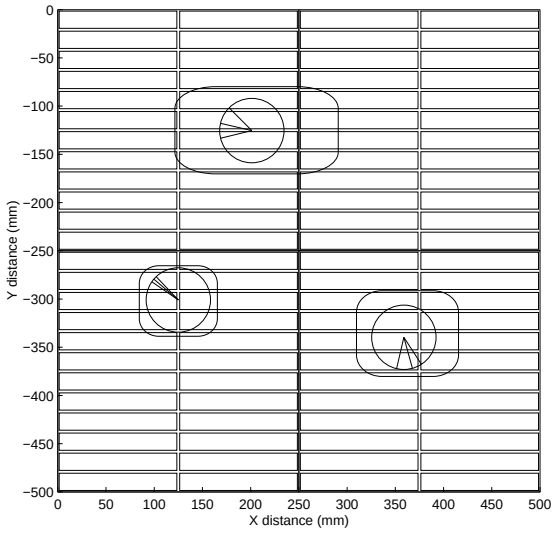

Fig. 8. Graphical location depiction for different positions

5 Improving Location Accuracy

After evaluating the location accuracy, ways of improving the performance were investigated. Since the investigated improvements are not implemented in the current prototype, they were verified with further simulations.

5.1 Possible Improvements

The handshaking protocol was designed with the aim of a fast connection process. As a result, the information included in the pad mapping was kept to a minimum. In particular, only one pad per link was recorded, and possible additional links were not reported at all. However, for the purpose of achieving a more accurate location, a more detailed pad mapping is desirable.

The amount of information provided in the pad mapping can be improved in three ways. The first improvement *"Using Duplicate Pads"* includes information about all object pads touching each of the strips used. The second improvement *"Using All Links"* includes information on all strips the object might touch, and not just the strips used for connection. The two improvements were also combined as *"Using Both Optimisations,"* effectively incorporating information about every object pad which makes contact with a surface strip.

5.2 Evaluation of Improvements

A second set of simulations was carried out to evaluate the effects of the improvements. Again, the number of required links was varied from two to six, and each configuration

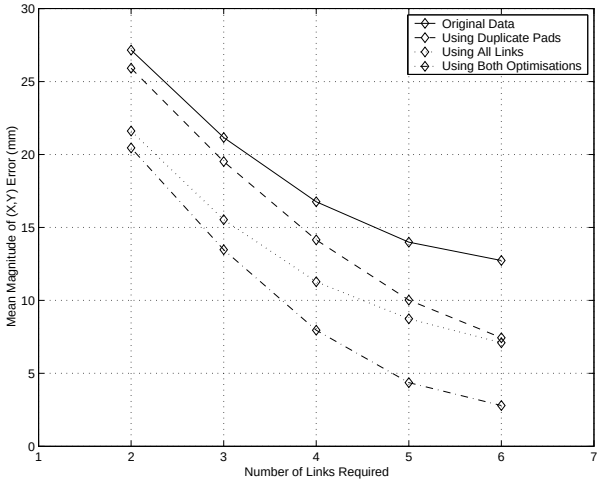

Fig. 9. Mean magnitude of positional error in location simulations using improved data

was tested at one million randomly chosen locations. The results, for the variables of the magnitude of (x, y) and θ, are shown in Figs. 9 through 12.

As the graphs show, both optimisations independently improve mean accuracy in both variables, and when using both optimisations the mean improvement made is approximately additive. This is because the optimisations are highly independent, as one only operates on tile pads already in use, whereas the other refers to unused tile pads only. For the maximum error, the use of both optimisations results in greater-than-additive gains in accuracy. This is again explained by the independent nature of the optimisations, which means that their worst case error occur in different situations.

For the magnitude of the maximum positional errors shown in Fig. 10, the duplicate pads optimisation is seen to give no improvement. This shows that the worst case for this optimisation coincides with that for the unmodified case. It is conjectured (though not proven) that these worst case errors happen when an object spans two columns, but pads in only one of the columns are used; in this case, both cases would incur a large x error.

Figure 11 shows the duplicate pads optimisation to be more effective for θ; this can be explained by simply noting that more additional pad mappings are provided in the duplicate pads optimisation. This is particularly true of cases where no extra links can be found, and for these cases the duplicate pads data is likely to be particularly plentiful (as all the object's pads are divided amongst a smaller number of tile pads).

Finally, the example of the four link object used previously is revisited. When using both optimisations, the average precision of the location data is increased from 1.6cm to 0.8cm, with the worst case reduced from 6.1cm to 3.1cm. For orientation, the average error is improved from 10° to 1.4°, and the worst case from 41° to 8.5°. This level of accuracy makes Networked Surfaces suitable for providing precise positions and orientation data to many applications.

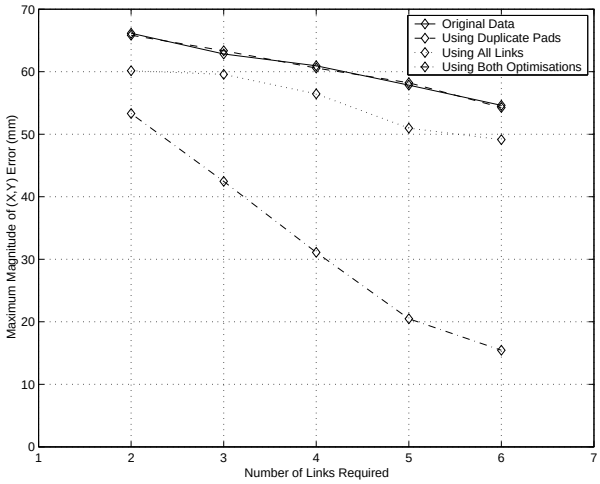

Fig. 10. Maximum magnitude of positional error in location simulations using improved data

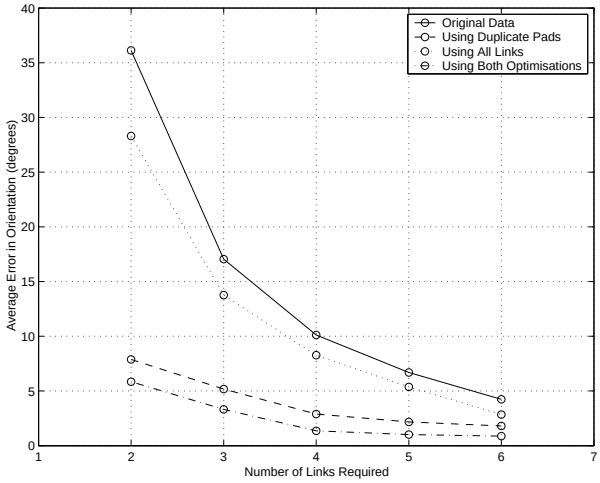

Fig. 11. Mean orientation error in location simulations using improved data

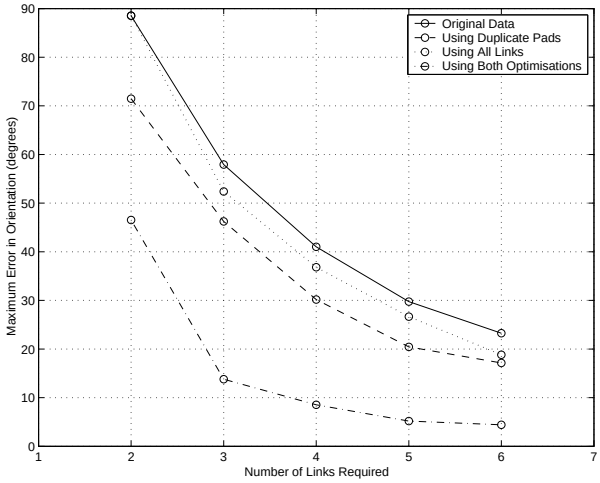

Fig. 12. Maximum orientation error in location simulations using improved data

6 Comparison to Other Location Systems

In order to put the performance of the Networked Surface as a location system into perspective, the accuracy of the location information it produces has been compared to a variety of other indoor location systems.

6.1 Location Systems

Location systems can be grouped by the type of technology they use. Systems from each of the prominent groups are reviewed below, and later compared to location using Networked Surfaces.

Infrared-Based Location Systems. The Active Badge system [23, 8] uses small badges that periodically send infrared pulses containing a unique identification number. The badges can be worn by users or they can be attached to equipment like workstations. Receivers in a room record the sightings of the infrared pulses and pass them on to a central server which converts the readings into a location. The system's resolution is of room-scale granularity.

The Locust Swarm [22] system is another infrared-based system. In this system the beaconing transmitters are located in the environment, and the receivers are located on the tagged objects. This configuration allows for privacy guarantees to be made, since the objects infer their own location without the need for a central server.

The ParcTab system [20] explores how devices like Personal Digital Assistants can be enhanced to include location-aware applications. The infrared based location technology gives room-scale accuracy. The system also provides networking capabilities with a bandwidth of about 20 kbit/s using infrared.

Ultrasonic Location Systems. The Active Bat system [24] uses a grid of fixed receivers to detect ultrasonic pulses emitted by small badges which can be carried by users or attached to devices. The transmitters and receivers are synchronised using RF pulses, and the times of flight of the ultrasonic pulses to many receivers are used to calculate the transmitter's position to within 3cm. For orientation, two transmitters can be used on an object, or the reception pattern from a single transmitter.

The Cricket system [17] also uses a combination of RF and ultrasonic pulses. "Beacons" placed in the environment periodically send their location as a radio message, and simultaneously emit a short ultrasonic pulse. "Listeners," which are attached to devices or carried by users, measure the time of flight of the ultrasonic pulse to estimate the distance to the beacon. The system can estimate postitions to within two metres, and by using multiple receivers can determine orientation to an accuracy of 3°.

The Dolphin system [9] is similar to the Active Bat system, but uses broadband ultrasonic transmissions. This allows it to achieve robustness in the presence of ultrasonic noise, and to allow multiple transmitters to send simultaneously, which are properties not shared by the two narrowband systems described above. The system is accurate to 2cm, and could provide orientation similarly to the Bat system.

Vision-Based Location Systems. The TRIP System [14] uses cameras to recognise circular barcodes, identifying both the pattern on the barcode and the location and orientation with respect to the camera. It has the advantage of using very cheap tags (printed paper), but tags are also easily obscured. The system is fairly accurate, with errors of about 3cm and 1° at a range of 150cm.

The Easyliving project's person tracking system [13] detects location by using stereo cameras to track individuals as they walk around a room. Colour histograms are used to keep track of each user separately. The system locates users with 10cm accuracy along the 2D plane of the floor.

RF-Based Location Systems. The RADAR project [4] uses signal strength readings of WaveLAN cards received at multiple well known locations to gain an estimate for the position of the card. The location of devices can be estimated to within three metres.

The "Nibble" system [7] also uses WaveLAN signal strength, but with the prototype system having far more wireless access points (10–14 in one experiment). It is therefore able to achieve high accuracies for room-grain location (97% in that case).

Another example for using an RF network to locate devices is described in [12]. In this project a grid of Bluetooth transceiver nodes installed in a trade show hall was used to provide networking. In addition, the system was capable of tracking the position of devices, to an accuracy of 10–20m.

Systems that can generate location information based on using wireless network technology are similar to the Networked Surfaces project, in the sense that they use a technology primarily designed for networking to provide location information as well.

6.2 Comparison

Table 2 summarises the location technologies discussed and compares them against the properties of Networked Surface location.

Table 2. Comparison of indoor location technologies

System Name	Medium Used	Typical Accuracy	Orientation Provided	Networking Provided
Active Badge	Infrared	Room-grain	No	No
Locust Swarm	Infrared	Room-grain	No	No
ParcTab	Infrared	Room-grain	No	20kbit/s
Active Bat	Ultrasonic	8cm	Yes	No
Cricket	Ultrasonic	2m	Yes	No
Dolphin	Ultrasonic	2cm	Yes	No
TRIP	Vision	3cm	Yes	No
EasyLiving	Vision	10cm	No	No
RADAR	WaveLAN signal strength	3m	No	2Mbit/s
Nibble	WaveLAN signal strength	Room-grain	No	11Mbit/s
Bluetooth-based	Nearest base station	10–20m	No	1Mbit/s
Networked Surface	Electronic	2cm	Yes	5Mbit/s

As the table shows, the Networked Surface provides the most accurate location service out of the systems described, and is the only system to provide both orientation and networking as well. In addition, the Networked Surface has been shown to provide locations with a bounded maximum error, and to provide locations with 100% reliability for connected objects, whereas other systems have unbounded errors, and fail to provide locations some of the time.

The disadvantages of Networked Surface location lie in extensive hardware requirements, the fact that only certain classes of objects can be located, and also that devices can only be located while they are connected to a Surface. In addition, the location of the Surface itself must be found using another means; in the simplest case this may be statically defined.

7 Uses of Networked Surface Location

Interactions between Networked Surfaces location and other context-aware systems and applications are discussed below.

7.1 Interaction with Other Location Systems

The Networked Surface can only provide location data for electronic devices which are commonly placed on surfaces. Many classes of device are therefore unlocateable. However, it is possible to use other location systems in tandem with Networked Surfaces, for locating devices which are not surface-based, or for keeping track of objects even when they are not on a Surface.

One useful synergy might be found with the TRIP system. TRIP is able to locate non-networked objects, placed anywhere in an environment. However, TRIP relies on

having networked cameras at known locations. For these cameras, a Networked Surface may be considered an ideal infrastructure, as it provides both convenient mobile networking and accurate location. The systems therefore complement each other well.

7.2 Context-Aware Middleware

Instead of implementing applications independently for each location system, it makes sense to use a mediating layer, which takes location data from one or more systems and presents it in a generic fashion to applications. Such a layer might described as "context-aware middleware." Described below are a few context-aware middlewares which have a presence at the LCE, making integration with Networked Surfaces a feasible future work.

The SPIRIT project [3] currently collates location information from Active Badges and Active Bats. There is a basic notion of conflict-resolution (Bat readings always overrule Badge readings). The system is built using CORBA interfaces; integration of Networked Surfaces location would therefore involve using CORBA calls when objects connect and disconnect.

Another such framework is found in the QoS DREAM project [15], which looks at the provision of Quality of Service for distributed multimedia applications. Location information is used to facilitate redirection of multimedia streams to follow users. The system currently uses the Active Badge as a location mechanism. Due to its event-driven architecture, integration with the Networked Surface may be achieved by mapping object connection and disconnection onto QoS DREAM events.

7.3 Applications

By using context-aware middleware as described above, a whole range of location-aware applications can be made to use Networked Surface location information. A short survey of existing applications is presented here, as well as a few novel context-aware applications specific to Networked Surfaces.

Firstly, visualisation applications have been developed to display location information to interested users in text [8] or graphical [2] form. Next, "follow-me" applications allow users to work while moving around, an example being "desktop teleporting" [19]. Thirdly, applications can be made location-dependent, e.g. the Stick-E note system [5] and the Cyberguide project [1]. Finally, "intelligent environments" can be constructed, including interactive desks [16] and walls [18], offices [2], and home environments [6].

In addition to the existing applications above, Networked Surfaces can lend themselves to new types of application, in which both the networking and location-aware aspects of the technology are used. While such applications are not the topic of this paper (see [21]), two applications are outlined below.

Firstly, Networked Surfaces may be used to automatically configure connections between devices, based on their physical placement. For example, a peripheral (e.g. a keyboard or digital camera) placed on a Surface may be connected to the closest computer. Data transfers might also be automatically inferred, e.g. placing a PDA next to a keyboard might cause synchronisation of data between the computer and PDA.

A second application can be found in the use of Surfaces for human-computer interaction, similarly to the intelligent environments described above. Input can be achieved using the placement and movement of devices in special ways. Output could be provided for with the output facilities of devices on the Surface, or by building outputs such as LED's into the Surface. Example applications include status displays (e.g. visual representations of data flow such as during synchronisation of a PDA), or notifications (e.g. an email icon appearing and moving towards the closest computer).

8 Conclusions

In this paper it was shown that Networked Surfaces can successfully be used to locate devices. The chosen location algorithm was evaluated with both simulations and measurements. Methods of improving the location accuracy were then discussed, and verified using simulations.

Using these improvements, the location of objects placed on a Networked Surface can be given with a mean error of 8mm and a guaranteed maximum error of 32mm for position, and 2° (mean error) and 9° (max. error) for orientation. These results show that the system's performance is better than that of many dedicated location systems. Also, locations are provided with 100% reliability, and with a bounded maximum error.

Acknowledgements

The authors would like to thank Mike Addlesee, Glenford Mapp, and Andy Hopper for their guidance and helpful suggestions. Thanks are also due to Michael Scott for his insightful comments concerning the location algorithm, and Rob Harle, Eli Katsiri and Diego López de Ipiña for informative discussions about integration with their research.

Frank Hoffmann's PhD research was funded by AT&T Laboratories Cambridge.

James Scott's PhD research was funded by the Schiff Foundation of Cambridge, and by AT&T Laboratories Cambridge.

References

[1] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks*, 3(5):421–433, 1997.

[2] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a Sentient Computing system. *IEEE Computer*, 34(8):50–56, August 2001.

[3] Noha Adly, Pete Steggles, and Andy Harter. SPIRIT: a resource database for mobile users. In *Proceedings of CHI '97*. ACM, March 1997.

[4] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of INFOCOM 2000*, volume 2, pages 775–784. IEEE, March 2000.

[5] Peter J. Brown, John D. Bovey, and Xian Chen. Context-aware applications: From the laboratory to the marketplace. *IEEE Personal Communications*, 4(5):58–64, October 1997.

[6] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven Shafer. EasyLiving: Technologies for intelligent environments. In *Proceedings of HUC 2000*, pages 12–29, September 2000.

[7] Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz. A probabilistic location service for wireless network environments. In *Proceedings of Ubicomp 2001*, pages 18–34, September 2001.

[8] Andy Harter and Andy Hopper. A distributed location system for the Active Office. *IEEE Network*, 8(1):62–70, January 1994.

[9] Mike Hazas and Andy Ward. A novel broadband ultrasonic location system. In *Proceedings of Ubicomp 2002*, September 2002.

[10] Frank Hoffmann, James Scott, Mike Addlesee, Glenford Mapp, and Andy Hopper. Data Transport on the Networked Surface. In *Proceedings of LCN 2001*. IEEE, November 2001.

[11] Andy Hopper. The Clifford Paterson Lecture, 1999. Sentient Computing. *Philosophical Transactions of The Royal Society of London*, 358(1773):2349–2358, August 2000.

[12] Rolf Kraemer. Bluetooth Based Wireless Internet Applications for Indoor Hotspots: Experience of a Successful Experiment During CeBIT 2001. In *Proceedings of LCN 2001*, pages 518–524. IEEE, November 2001.

[13] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for EasyLiving. In *Proceedings of the Third IEEE International Workshop on Visual Surveillance*, 2000.

[14] Diego López de Ipiña. Video-based sensing for wide deployment of sentient spaces. In *Proceedings of PACT 2001*. ACM/IEEE, September 2001.

[15] Hani Naguib and George Coulouris. Location information management. In *Proceedings of Ubicomp 2001*, pages 35–41. ACM, September 2001.

[16] William Newman and Pierre Wellner. A desk supporting computer-based interaction with paper documents. In *Proceedings of CHI '92*. ACM, May 1992.

[17] Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth Teller. The Cricket Compass for context-aware mobile applications. In *Proceedings of MobiCom 2001*. ACM/IEEE, July 2001.

[18] Jun Rekimoto and Masanori Saitoh. Augmented Surfaces: A spatially continuous work space for hybrid computing environments. In *Proceedings of CHI '99*. ACM, May 1999.

[19] Tristan Richardson, Frazer Bennett, Glenford Mapp, and Andy Hopper. Teleporting in an X window system environment. *IEEE Personal Communications*, 1(3):6–12, 1994.

[20] Bill N. Schilit, Norman Adams, Rich Gold, Michael Tso, and Roy Want. The ParcTab mobile computing system. In *Proceedings of WWOS-IV*, pages 34–39. IEEE, October 1993.

[21] James Scott, Frank Hoffmann, Glenford Mapp, Michael D. Addlesee, and Andy Hopper. Networked Surfaces: A new concept in mobile networking. *ACM Mobile Networks and Applications*, 7(5), October 2002.

[22] Thad Starner, Dana Kirsh, and Solomon Assefa. The Locust Swarm: An environmentally-powered, networkless location and messaging system. In *Proceedings of ISWC '97*. IEEE, 1997.

[23] Roy Want, Andy Hopper, Veronica Falcao, and Jon Gibbons. The Active Badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

[24] Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the Active Office. *IEEE Personal Communications*, 4(5):42–47, October 1997.